SESSION

TEACHING PROGRAMMING + SOFTWARE ENGINEERING + PERFORMANCE ENHANCEMENT METHODS + INTEGRATED LEARNING AND RELATED ISSUES

Chair(s)

TBA

Mining Student Time Management Patterns in Programming Projects[1]

Dr. Dale E. Parson and Allison Seidel
Kutztown University of Pennsylvania
Kutztown, PA, 19530, USA

Abstract

Computer science faculty members cite procrastination as one of the key causes of poor student performance in programming projects. In contrast, students cite conflicting demands for time. This study uses a tool-driven process of automated compilation and testing of student programs to collect student-project data. Data include when, for how long, how often, and with what magnitude of effort and accomplishment, students engage in work to complete programming assignments. Participation is voluntary, and data from auxiliary sources, including a questionnaire on conflicting demands on time, complement automatically collected data. Analyses reveal that procrastination and excessively brief work sessions are the main indicators of problems for students with inadequate prior success in earlier computer science courses. Some students with successful track records know when they can afford late starts and short sessions. The time of day that students work is a contributing factor to success. The goal is to build an automated warning system for at-risk students.

Keywords: data mining, programming assessment, student programming, time management.

1. Introduction

The project reported in this paper grew out of an initial offering of a master's level course in data mining using the Weka toolset [1,2] in conjunction with the annual offering of two sections of the Java Programming course in spring 2013. Graduate students offered advice on mining the project work habits of undergraduates, and Java students voluntarily supplied the initial data set. Java Programming at Kutztown University is an elective major course that includes sophomores, juniors and seniors after they complete the CS1-CS2 introductory course sequence using C++. The authors extended data collection to two sections of undergraduate Operating Systems in fall 2013, and to two sections of Java Programming and one section of Programming Languages in spring 2014. The range of courses and student experience levels help to distinguish consistent attributes from incidental ones in predicting student project success as a function of time management and other student work related data. None of the courses include in-class programming time. All project work takes place as homework.

The initial investigation grew out of an interest in quantifying the folklore on both sides of the faculty / student divide. Faculty members cite procrastination as a primary cause of poor results in student programming projects. Students cite conflicting demands on their time from projects and exams in other courses. While both factors contribute to project success to some degree, their contributions are far from simple and linear.

A goal is construction of an advisement program that would warn at-risk students when their patterns of work on programming projects begin to exhibit signs of problems. Anticipated use of this program by students would be voluntary, as is participation in the ongoing study itself.

The authors chose to report preliminary results before completion of the full study because we have uncovered useful information that we are communicating to students in current courses. Encouraging students to avoid pitfalls identified in preliminary results may help to uncover the effectiveness of passing this information along to students.

2. Related work

Edwards, et. al. have previously reported late starts in programming projects as clearly associated with poorer results on such projects [3]. That study utilized data from three programming courses for over five years, in contrast to the present preliminary study. It confirmed results from earlier studies about the correlation of earlier project starts with better rates of project success. Unlike the current project, that study eliminated both consistently well-performing students and consistently poorly-performing students from the analysis in order to focus on intra-student attributes that vary between successful and unsuccessful

[1] This study was supported in part by a Kutztown University Assessment Grant in 2013.

projects. The current study seeks to uncover consistent work habits leading to success and failure, regardless of individual student correlations. In fact, the present study is particularly interested in detecting patterns within at-risk students, including both consistently poorly-performing students and intermittently poorly-performing students. The cited study considered only the latter population.

Results of the present study agree with the results of the cited study, while detecting additional significant attributes of student work habits that alter those basic results.

Edwards and Ly have reported on automating analysis of the specific types of problems that occur in running student programs [4]. That analysis differs in nature from the current project, which focuses on correlations between student work patterns and project success or failure, as contrasted with specific types of project failure.

Mierle, et. al. examined student code repositories, based on file modifications that appear in file change submissions and log files [5]. They found a weak correlation between normalized number of lines of code per revision and student success in terms of final grade, and no correlation with timing of student work. The present study differs in nature by logging and analyzing student work at a much finer temporal grain, that of individual *make* actions within each student's private workspace. The present study uncovers more detailed correlations.

The most recent related study examined confirms the correlation of poor programming project results with late project starts [6]. That study allowed student submission of programs to additional, opaque automated tests (so-called *release tokens*). It concluded that availability of these tests might discourage students from writing their own test cases, and might encourage procrastination because students can count on additional available tests no matter how late they start. The current project takes a different approach to testing, modeled after the industrial experience of the instructor. The instructor supplies test cases, requires students to write additional test cases for some projects, and uses additional test cases not available to the students for grading. The latter sets of tests emulate customer *acceptance testing* not available to software providers. The present study collects data on successful and unsuccessful test runs for instructor-supplied and student-required testing, but it does not use a limited number of opaque test runs as a variable. The current study finds several attributes that correlate with project success, in addition to start time.

3. Data Collection & Extraction

3.1 Using *make* for attribute collection

Most of the programming projects studied use the GNU *make* utility [7] on a Unix server for compilation, testing, and submission of student programs. A student types *make test* in a project directory, leading to automatic compilation when object or executable files predate their corresponding source files, followed by automated tests supplied by the instructor, and then by automated tests written by students when required. Automated submission of completed programs takes place via the *make turnitin* command. For projects where students offload code from the Unix server to laptops for development of graphical user interfaces (GUIs), the instructor supplies an executable Java archive that serves as an alternative, project-specific makefile.

Using either mechanism, the makefile captures and compresses data from the project directory and archives it for later analysis. Analysis defers until after course completion and grading as part of the agreement between the instructor and students. Data collected for the study play no role in grading. A student earns one bonus point on a scale of 100 project points for participating in the automated study, and an additional point for completing a short survey explained in the next section. Students can also earn these two points by performing actions that remove their data from the study. Institutional ethical standards preclude giving bonus point incentives only to participating students. Participation is voluntary, and students must not feel grade pressure to participate.

Listing 1 shows the first category of collected data, the actual zip archives containing detailed student information. An archive file name encodes the student identifier (obscured here), the date and time of the triggering *make* action, and an identifier for that action. The first two BUILD lines of Listing 1 show two failed attempts to compile the assignment, followed by a BUILD leading to a BUILT record, signifying successful compilation. The TESTING line signifies the start of automated tests, and the TESTED line signifies successful execution of tests. Typically, the student triggers compilation, and testing when compilation succeeds, by invoking *make test*, which triggers *make build* for updated source files. It is possible to determine the time and frequency of a student's attempts at compilation and testing, and the success of these attempts, simply by decoding the archive file names. Data logging activities remain invisible in the interest of minimizing impact on student workflow.

```
idN_2014-02-12-12-37-43-EST_BUILD.zip
idN_2014-02-12-12-38-48-EST_BUILD.zip
idN_2014-02-12-12-41-50-EST_BUILD.zip
idN_2014-02-12-12-41-53-EST_BUILT.zip
idN_2014-02-12-12-41-53-EST_TESTING.zip
idN_2014-02-12-12-41-55-EST_TESTED.zip
```

Listing 1: Archives of student *make* data

Each archive contains two files. Listing 2 shows an example Unix listing file extracted from an archive. It is a listing of the student's project directory at the time of invoking make. Each line shows student ID, the number of bytes in the file, the file's most recent modification date and time, and the file's name. By comparing file sizes and modification timestamps from successive archives, it is possible to determine which files have changed, and the net change in byte size for those files, since the previous make action for that student project.

The other file extracted from each archive of Listing 1 contains the contents of multiple text files. Each logged make action concatenates all text files of interest in the project, in this case Java source files, into a single file with markers giving start of each file, file name, logging time, and file contents. The initial use for collecting source files has been to use the Unix *diff* utility to determine number of lines *added*, *changed*, and *deleted* for each source file since the previous make action. These files could also support analysis of types of student solution and error mechanisms in a subsequent study.

```
TESTED 2013-03-24-21-45-38-EDT
~idN/JavaLang/FillWord2
idN     5836 Mar  3 15:43         FillWordTest.java
idN     57 Mar  3 16:01           testjava.txt
idN     6248 Mar  3 16:01         testjava.ref
idN     70 Mar  3 16:06           testjava2.txt
idN     2037 Mar  3 16:07         makefile
idN     8951 Mar  3 16:07         testjava2.ref
idN     1183 Mar 13 17:33         IFillWord.java
idN     3474 Mar 14 12:59 FillWordGrows.java
idN     8849 Mar 24 21:33 FillWordBasic.java
idN     5027 Mar 24 21:37 FillWordHelper.java
```

Listing 2: Listing of student project directory

In summary, archiving collects all available project data for a student every time a student compiles, tests, or turns in a project, including the success or failure of compilation and testing. Since compilation and testing are automated, it is possible to determine, automatically, exactly which files fail compilation and which tests fail testing. For the current study the focus is on time management patterns, the magnitude of changes per work session, time of day of each session, and related data discussed in the next section.

3.2 Collection of auxiliary attributes

There is no means for automatically collecting data about conflicting demands on student time. The study uses a survey with three questions: How many computer science projects from other courses were given out during the project period? How many computer science projects from other courses were due during the project period? How many exams from any course took place during the project period? This study does not measure non-course time conflicts such as jobs or extracurricular activities.

Since responses are subjective and accuracy of the answers is less reliable than the makefile-collected data of the previous section, the study treats survey answers as student perceptions of these potential sources of time conflicts. Correlation of student perceptions with project success is a measure that the study can inspect objectively.

Student data from other sources include *year* (freshman, sophomore, etc.), *track* (the department has software development and information technology tracks for students), *course, semester, project number,* and *project start* and *end dates and times*. The instructor collects *student email questions* about projects, classifying them into the following categories: 1) informed and detailed, 2) uninformed and vague, and 3) the total of (1) and (2). Email provides one sample measure of student interaction with the instructor. This study does not measure classroom attendance or office hour meetings with the instructor.

So-called *target attributes* include project numeric and letter grade, final course numeric and letter grade, and centile ranking of project and course numeric grades. The latter measures are useful in distributing clumped grades during analysis. A script extracts these data from a grading spreadsheet after the semester ends.

3.3 Student cumulative grade data

The initial plan was not to use cumulative student grade data coming into the course. The intent was to base analysis strictly on conditions and actions relating directly to the projects. However, initial analysis determined that some students could perform well with seemingly bad works habits, while others could not. That fact may account for some of the lack of correlation between project start time and project success in the study by Mierle, et. al. [5]. Consequently, this study incorporates four additional attributes into the dataset: 1) cumulative grade point average in computer science courses, 2) number of credits in computer science courses, 3) overall cumulative grade point average for all courses, and 4) total credits earned. These four attributes are recorded at the start of the course. Analysis reveals that including these attributes helps to sort out students who can afford to use what might be considered "bad habits" from those who cannot. These attributes are helpful in identifying potentially at-risk students who could subscribe to an automated warning system derived from this study.

4. Data Analysis

4.1 Distinguishing attributes

The study uses 90 attributes concerning student project activities and related data discussed in the previous section. These attributes fit into the following categories.

1. **Student data** include ID, major track, and year in university.
2. **Course-project data** include course number, semester, project number, and project start and end date-time.

3. **Date-times for start and completion of student work** come in five forms: A) hours from handout until student start, B) hours from completion until project due, C) hours from student start until project due, D) hours from handout until student completion, and E) hours from student start until completion.
4. Other time management attributes relate to the concept of a *work session*, which for this study is a contiguous period of captured make actions with no intervals between actions >= 60 minutes in length. Attributes record the min, max, mean, sample standard deviation, median and mode for the following data: **length of session time** in minutes, and **time between sessions** in hours.
5. Additional temporal attributes are **total session time**, total **number of sessions**, and **session time-of-day**. Graduate students suggested the latter data category, which comes in six divisions: sessions centered between 12 and 3:59 AM, between 4 and 7:59 AM, between 8 and 11:59 AM, and their three afternoon-evening counterparts.
6. The study also analyzes size of work (min, max, mean, sample standard deviation, median and mode) for the following per-session data attributes: **number of source file bytes changed**, **number of source files modified**, number of source **lines added**, **lines deleted**, and **lines changed**.
7. Informed and detailed **email messages** to the instructor, uninformed and vague emails, and the total of these two classes comprise another class of data as previously discussed.
8. The *target attributes* of project numeric and letter grade, final course numeric and letter grade, and centile ranking of project and course numeric grades constitute the final category.

Analysis uses only one target attribute at a time because these attributes are partially redundant. Project numeric grade trivially determines project letter grade, for example, without considering other attributes. After some analysis the study dropped use of overall course grades because these course grades are a weighted average of course projects and exams, and some students are habitually poor test takers. Project preparation patterns showed no clear correlation to exam results. The study also dropped use of letter grades in favor of using binned (discretized) numeric grades in order to allow exploration of binning strategies. The primary target attribute is *project numeric grade*, with *project centile ranking* also considered in detail.

Data mining has two different modes of operation for investigators. The first is to *make data relationships clearer to human investigators*. The second is to support creation of *automated programs for pattern recognition and response*. The former is the mode of the current study. The latter is the mode for the anticipated automated early warning system for at-risk students.

The study used two complementary approaches to thin the 84 non-target attributes down to a set that exhibits correlation with target attributes in a way that makes the data relationships clearer to human investigators. The first was repeated manual application of Weka's OneR rule-based machine learning algorithm [1,2] that uses the minimum-error attribute for prediction of discrete numeric attribute bins. OneR selects the single most accurate non-target attribute to predict the target attribute. Our process was to use OneR repeatedly to select the next-most predictive attribute, then remove that attribute, and then repeat the process. This iterative process also eliminated partially redundant attributes, along with attributes for which there were significant numbers of missing values. For example, the initial study of the Java course did not capture source files for GUI development projects run on laptops, so it was necessary to discard diff-based line add/change/delete attributes when analyzing those projects.

The second approach was to use Weka's "Select attributes" capability, applying the "CfsSubsetEval" algorithm that evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them, in conjunction with a "BestFirst" search method that searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility. This is essentially the automated version of the semi-manual approach of the previous paragraph. Both yielded the same set of predictive attributes.

Jstr	hours from student start until due deadline
Mavg	average minutes of a work session
Mdev	sample standard deviation of Mavg
Snum	number of work sessions
Mtot	total minutes spent on the project
Cgpa	computer science GPA at semester start
GprjRank	centile ranking of project grade

If Jstr < 24.0 then 0 <= GprjRank <= 13.2
Elseif Jstr < 79.0 then GprjRank > 90.4
Elseif Jstr < 181.0 then 80.7 < GprjRank <= 90.4
Elseif Jstr >= 181.0 then GprjRank > 90.4
(36/111, 32.4%, instances correct)

Listing 3: OneR prediction from 6 attributes

Listing 3 shows Weka results for six of the most predictive attributes in predicting the centile ranking of the project grade. OneR shows that **Jstr**, the number of hours from the time the student started the project until the project deadline, is the single most predictive attribute. Other analyses duplicate this result. **Jstr** is more useful than the time from project handout until start, because project time periods vary by project, sometimes because of complexity, but also because of incidental reasons such as spring break or an exam in the course. **Jstr** is a measure of

Gprj project grade
If Cgpa < 2.24 then 0 <= Gprj <= 0.625
Elseif Cgpa < 2.635 then 0.625 < Gprj <= 0.815
Elseif Cgpa < 2.945 then 0.815 < Gprj <= 0.925
Elseif Cgpa >= 2.945 then 1.015 < Gprj <= 1.04
(28/111, 25.2% instances correct)

Listing 4: OneR prediction to project grade

procrastination because it shows how close to the project deadline a student has waited before starting.

Listing 3 shows that starting a project within the last 24 hours leads to the lowest centile bin, between 0 and 13.2 percentile. This observation accords with previous studies such as Edwards, et. al. [3], which states, "approximately two-thirds of the lower scores were received by individuals who started on the last day or later." However, the fact that OneR's predictions based on **Jstr** are correct only 32.4% of the time, and that one-third of the lower scores of [3] were received by individuals who started earlier than the last day, show that procrastination is not the only important attribute at work in determining project quality.

Listing 3 shows a problem with a simple monotonic interpretation of start time before deadline as the main attribute of interest. OneR predicts that students starting between 24 and 79 hours before the due date will attain the top centile bin (greater than 90.4 percentile), 79-to-181 hour starters will attain the next lower centile bin, and students starting >= 181 hours will attain the top bin. The problem is with the 24-79 group attaining the top bin. This result resonates with the instructor's experience that some students can start during the last few days, sometimes send very focused email questions that may not receive immediate responses, send subsequent "never mind, I found it" messages, and complete the project with good results. It is because of such students that the authors decided to include the computer science GPA and related attributes in the study. In fact, replacing centile ranking with project grade as the target attribute leads to the OneR rule of Listing 4, with computer science GPA (**Cgpa**) replacing **Jstr** as the most predictive attribute.

This fatalistic prediction of prior GPA as the most predictive attribute would be discouraging, were it not for the fact that this OneR rule is wrong 74.8% of the time for the Java Programming dataset. **Cgpa** is important, but its contribution is not deterministic. The next section takes up the detailed analysis used to find countervailing attributes.

4.2 Patterns for indication of success

The previous section deals with the search through available attributes. Lack of space precludes inclusion of strategies for the search through the space of available machine learning algorithms and correlation techniques. The guiding principle in this search was intelligibility. Algorithms and tools whose results are straightforward to interpret lead to the results reported in this section.

Table 1 shows the results of Simple K-means clustering [2] for 6 clusters, along with the full programming dataset, for the six most predictive attributes for project grade **Gprj** in Java Programming. Several relationships in the data appear at the surface.

First, the lowest average project grade of 57.19% in cluster 4 pairs with the latest starting time **Jstr** of 36.1875 hours before the project deadline. Cluster 4 is the smoking gun for the negative effects of procrastination.

However, cluster 0 with the second smallest **Jstr** of 57.875 pairs with a grade of 93.83%. Comparing these two clusters shows that the computer science GPA for cluster 0 is 3.3392 compared to 2.255 for cluster 4, suggesting that the cluster 0 students are better prepared for a somewhat late start; cluster 0's **Jstr** value is 60% greater than cluster 4's. There are other important differences. Cluster 0's average work session time in minutes **Mavg** is 68.5097, compared to cluster 4's 36.6927, and cluster 0's total session minutes **Mtot** is 174.125, compared to 4's **Mtot** of 100.6875. Cluster 0 students work in sessions that are 87% longer, and for overall time that is 75% greater, than cluster 4 students.

Cluster 5 with the second lowest **Gprj** value at 88.86% is significantly better than cluster 4 at 57.19%. The **Cgpa** at 2.6143 is also the second lowest, and **Mavg** is relatively low at 49.4403 minutes. However, cluster 5 students have a

attribute	full data 111 records	cluster 0 24 = 22%	cluster 1 9 = 8%	cluster 2 27 = 24%	cluster 3 7 = 6%	cluster 4 16 = 14%	cluster 5 28 = 25%
Jstr	167.9459	**57.875**	155.2222	249.7778	238	**36.1875**	245.25
Mavg	58.1449	**68.5097**	138.8896	42.4935	63.0163	**36.6927**	49.4403
Mdev	48.4055	36.8001	116.7501	44.7172	78.8199	21.9392	47.4618
Yavg	5153.661	4435.4986	20419.4735	2605.7539	5061.4645	5635.8594	3066.7782
Snum	5.4324	2.9167	4.4444	5.7407	14.1429	2.625	7.0357
Mtot	302.045	**174.125**	589.7778	242.2963	884.8571	**100.6875**	346.1786
Cgpa	3.055	**3.3392**	3.3411	3.6156	3.1429	**2.255**	2.6143
Gprj	90.01%	**93.83%**	99.56%	100.89%	102.29%	**57.19%**	88.86%

Table 1: Simple K-means clusters for the six most predictive attributes for project grade **Gprj** in Java Programming

high **Jstr** value of 245.25 hours, the second-highest number of sessions **Snum** of 7.0357, and a higher than average total session minutes **Mtot** of 346.1786. An early start and repeated application of time to the project pay off in giving cluster 5 students a **Gprj** that is 55% greater than that of cluster 4.

Finally, Table 1 shows that for most clusters, larger values for mean session minutes **Mavg** correlate with larger values for mean source bytes (characters) modified per session **Yavg**, but for cluster 4, the second highest **Yavg** of 5635.8594 bytes pairs with the smallest **Mavg** value. Cluster 4 students attempted, unsuccessfully, to complete a large amount of work in very little time.

Other algorithms such as the J48 decision tree and Bayesian inference [2] give results that are compatible with the K-means clusters, but that are harder to read and integrate into a paper because of the amount of detail in their logic. One additional example is the Pruned **M5P** model tree, which uses decision tree-like structure to select from among a set of linear regression formulas. Below is the M5P model tree for all Table 1 attributes except **Yavg**.

```
Cgpa <= 3.205 :
|  Jstr <= 66 :
|  |  Mavg <= 36.167 : LM1 (5/29.636%)
|  |  Mavg >  36.167 : LM2 (18/106.448%)
|  Jstr >  66 : LM3 (40/57.247%)
Cgpa >  3.205 : LM4 (48/31.383%)
LM num 1: Gprj =
   0.0012 * Jstr + 0.0009 * Mavg
   + 0.0214 * Snum - 0.0002 * Mtot
   + 0.2908 * Cgpa - 0.2251
LM num 2: Gprj =
   0.0001 * Jstr + 0.0009 * Mavg
   + 0.0214 * Snum - 0.0002 * Mtot
   + 0.2278 * Cgpa + 0.0515
LM num 3: Gprj =
   0.0001 * Jstr + 0.0007 * Mavg
   + 0.028 * Snum - 0.0001 * Mtot
   + 0.226 * Cgpa + 0.0893
LM num 4: Gprj =
   0.0008 * Mavg + 0.0143 * Snum
   - 0.0001 * Mtot + 0.1052 * Cgpa + 0.5199
```

Listing 5: Pruned M5P model tree

LM4: When the tree finds a **Cgpa** > 3.205, it goes to a linear formula with the highest constant value for the **Gprj** (51.99%) and the lowest weight for the **Cgpa**. These students are not typically at risk.

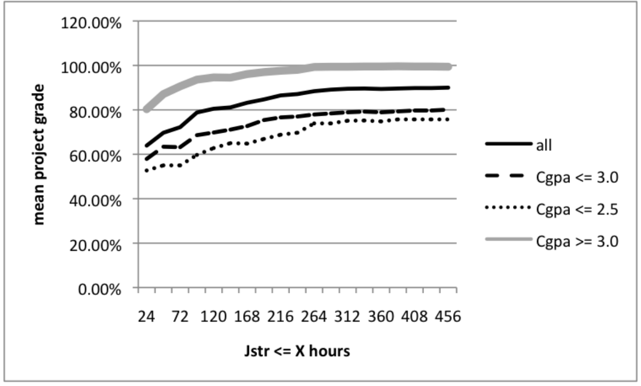
LM3: Otherwise, when **Cgpa** <= 3.205 and the start time **Jstr** is greater than 66 hours before the deadline, the tree goes to a linear formula with a much smaller constant value and greater dependence on the **Cgpa**.

LM2: With **Jstr** <= 66 and **Mavg** work time that exceeds 36.167 minutes, the tree goes to a formula with more weight on the **Cgpa** and a lower constant value.

LM1: Finally, with **Mavg** <= 36.167, the tree goes to a formula with the greatest dependence on **Cgpa** and a negative constant value. A low mean average session time is an automatic handicap.

There is nothing a student can do to change an incoming **Cgpa**, but the tree and formulas of Listing 5 indicate that students with lower **Jstr** and **Mavg** values correlate with lower constant values and more deterministic ties to the **Cgpa** within linear predictors, than students with higher **Jstr** and **Mavg** values. At-risk students need to start earlier and engage in sessions that exceed 36 minutes. Apparently, 36 minutes are not enough to get properly immersed in software development. Other Weka models suggest that **Mavg** should exceed 60 to 75 minutes.

M5P is one of the better predictors applied to these attributes, predicting about 55% of the test cases correctly for this dataset, with a mean absolute error of 13.04% on a **Gprj** scale of 100%. Predicting incorrectly 45% of the time, with an average miss of more than a 10% letter grade, is not great prediction. A student is a hard nut to crack. There are, seemingly, hidden variables not measured by the study. However, **Cgpa**, **Jstr**, and **Mavg** have strong correlation with project grade, and the latter two attributes are available for change by students.

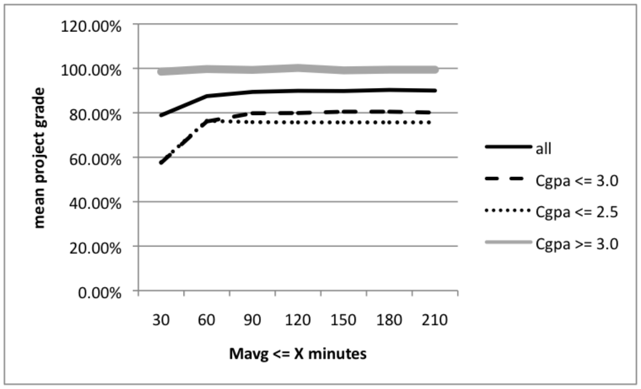


Graph 1: **Gprj** as a function of cumulative **Jstr** ranges

Graph 1 shows average relationships between **Jstr** value ranges and mean project grades **Gprj** for students in various **Cgpa** ranges. **Jstr** values to the right in the graph subsume **Jstr** values to the left. For example, 72 on the X axis means **Jstr** <= 72, a range that includes **Jstr** <= 24. The maximum grade of 127% in the dataset includes optional bonus points for one project. The authors computed Graph 1 by using the Weka preprocessor as a query tool to eliminate from the dataset **Cgpa** and **Jstr** values outside a desired range, and then entered the resulting mean **Gprj** value into an Excel spreadsheet to plot the graph. All four **Cgpa** curves show a minimum spread of 20%, i.e., two letter grades, across the range of

Jstr values. The graph cannot capture the nonlinear relationships of **Jstr** to **Cpga** illustrated by Table 1, but nevertheless it shows the overall importance of **Jstr**. All four curves level out at about 10 to 11 days. It is the instructor's experience that most students do not utilize project periods greater than two weeks in length very well, requiring the instructor to break big projects in a number of smaller assignments, each fitting in a two-week interval. Graph 1 confirms the validity of this practice.

Graph 2, also captured using Weka's value range filtering and Excel, shows related curves for the relationship of cumulative session minutes **Mavg** in relation to mean project grade **Gprj**. Work sessions less than 60 minutes in length lead to problems for most students, and sessions at least 75 minutes in length are better for at-risk students.



Graph 2: **Gprj** as a function of cumulative **Mavg** ranges

4.3 Secondary & problematic patterns

Weka's "Select attributes" capability suggests possible importance for work sessions with midpoints between 4 AM and 11:59 AM, i.e., morning sessions. Students in higher **Gprj** ranges did tend to work in the mornings, but they also worked more in the afternoons, took an evening break, and worked again during the four hours before midnight. At-risk students fell into two groups. One group started within the last 36 hours, leaving themselves no opportunity for time management. They did not work in the morning. The other group started well in advance of the deadline, performing most of their work during the four hours before midnight. Our conclusions are that some morning work after a night's rest helps to improve a project's quality, and that consistently working only at day's end, when a student is tired, leads to poor quality.

No other attributes correlate strongly with project success, including student survey data about conflicting demands on time. Also, the data collected in the fall 2013 Operating Systems classes were not useful because student projects consisted of modifying state machines drafted by the instructor that emphasized analysis and design, with very little coding, similar to solving a proof. Most of the work was in the pre-coding stage, and therefore not visible to data collection via make. Three spring 2014 programming courses will provide much more useful data.

5. Conclusions and Future Work

Procrastination is a culprit in poor results for programming projects, but it is not a simple, linear one. Some students who have performed well in past computer science courses apparently know their limits. However, for many students, and especially for at-risk students with lower computer science grade point averages at the start of a programming course, starting at least 11 days before the due date of a two-week project yields demonstrable benefits. Also, the minimum length of a programming work session should be at least 60 minutes, and preferably 75. Time of day of work is a contributing factor. In addition to starting early in the project cycle, students should not work solely at the end of long days, when they are tired.

There are additional data available for mining. Data collection in spring 2014 will more than double the size of the dataset. The M5P model tree algorithm promises to provide an accurate basis for constructing an early warning system for at-risk students. The plan is to implement a prototype system after completing analysis in summer 2014, with voluntary use by students to follow.

References

[1] Machine Learning Group at the University of Waikato, "Weka 3: Data Mining Software in Java", http://www.cs.waikato.ac.nz/ml/weka/, May 2014.

[2] Witten, Frank and Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Third Edition, Morgan Kaufmann, 2011.

[3] Edwards, Snyder, Pérez-Quiñones, Allevato, Kim and Tretola, "Comparing Effective and Ineffective Behaviors of Student Programmers", *Proceedings of ICER '09: International Computing Education Research Workshop*, Berkeley, CA, August 2009.

[4] Edwards and Ly, "Mining the Data in Programming Assignments for Educational Research", *Proceedings of the International Conference on Education and Information Systems: Technologies and Applications* (EISTA'07), Orlando, FL, July 12-15, 2007.

[5] Mierle, Laven, Roweis and Wilson, "Mining Student CVS Repositories for Performance Indicators", *Proceedings of the 2005 International Workshop on Mining Software Repositories*, St. Louis, May, 2005.

[6] Spacco, Fossati, Stamper and Rivers, "Towards Improving Programming Habits to Create Better Computer Science Course Outcomes", *Proceedings of ITiCSE 2013, the 18th Annual Conference on Innovation and Technology in Computer Science Education*, Canterbury, UK, July 1-3, 2013.

[7] Free Software Foundation, *GNU Make* home page, http://www.gnu.org/software/make/, May 2014.

Introducing Java's Enumerated Type

J. Mohr
University of Alberta, Augustana Campus,
4901 46 Ave., Camrose, Alberta, Canada T4V 2R3
jmohr@ualberta.ca

Abstract—*The enumerated types that were introduced in Java 5.0 provide a way to treat arithmetic operators as enumeration constants with an* `eval` *method that is customized for each operator, allowing us to take an object-oriented approach to applying an operator to its arguments, using dynamic dispatch instead of case logic, when implementing a calculator or an arithmetic expression evaluator. When used in conjunction with variable arity methods (or* `varargs`*, also introduced in Java 5.0) or by passing an operand stack as the single argument to an operator's* `eval` *method, an enumeration for operators can be extended to handle operators of differing arities using a single abstract* `eval` *method in the enumerated type. Such an example gives an opportunity to expose our students to Java's version of enumerated types by using an* `Operator` *enumeration in an assignment on infix expression parsing and evaluation.*

Keywords: Enumerated types, enum, expression evaluation, Java, variable arity parameters

1. Introduction

Arithmetic expression evaluation is a topic of study in many data structures or algorithms courses and is covered in several data structures and a few algorithms textbooks. In some cases, the evaluation of postfix expressions is introduced as part of the discussion of stacks [1]–[5]; in others, the evaluation of expression trees is discussed when binary trees are introduced [1]–[3], [6]–[8]. Some textbooks present the subject of infix-to-postfix conversion in conjunction with expression evaluation [3], [6], [9], [10]. Kruse and Ryba [11] dedicate a chapter (Ch. 13, pp. 594–645) to a case study on Polish notation, including the evaluation of prefix and postfix expressions and the translation from infix form to reverse Polish (postfix) form. Smith [5] shows how to combine infix-to-postfix translation with postfix expression evaluation to evaluate infix expressions using two stacks; Koffman and Wolfgang [6] present the same algorithm in an end-of-chapter programming project.

Most textbooks that treat the topic of arithmetic expression evaluation confine their coverage to the evaluation of binary operators, which is especially appropriate if the topic is presented as part of the discussion of binary trees. Those that discuss the conversion of infix expressions to their postfix equivalents usually include the handling of parentheses and operator priorities, but there is typically no discussion of associativity or of operators with arity other than binary. Of the textbooks cited above, only Weiss [10] and Kruse and Ryba [11] discuss right associativity (the right-to-left evaluation of operators like exponentiation), and only the latter textbook deals with both unary and binary operators.

Kruse and Ryba are also unique in their use of an enumeration for token types, including `enum` constants such as `operand, unaryop, binaryop, rightunaryop, leftparen` and `rightparen`.[1]

The introduction of enumerated types in Java 5.0 (release 1.5) provided the opportunity to treat arithmetic operators as enumeration constants with an `apply` or `eval` method that is customized for each operator. This allows us to take an object-oriented approach to applying an operator to its arguments, using dynamic dispatch instead of case logic through *constant-specific method implementations* [12, p. 152]. This also provides a good opportunity to expose our students to Java's version of enumerated types in conjunction with the discussion of arithmetic expression parsing and evaluation.

2. Enumerated Types

Niklaus Wirth introduced the enumerated type in Pascal (1970) as a user-defined ordinal type that "specifies an ordered set of values by enumerating the constant identifiers which denote the values" [13, Ch. 5]. It associated the first constant listed with the value 0; the second, with 1; and so on. It provided a high-level, application-oriented binding construct [14] that improved program readability and reduced errors by replacing ad hoc constructs such as

```
const
  Mon = 0;
  Tue = 1;
    . . .
  Sun = 6;
```

with

```
type Day = (Mon, Tue, Wed, Thu, Fri,
            Sat, Sun);
```

[1] Lambert and Osborne [3] give an example of a parser that uses a `switch` statement with case labels such as `PLUS, MINUS, MUL, DIV` and `L_PAR`, but these are implemented as defined constants rather than `enum` constants, an ad hoc approach that will be discussed below as the *`int` enum pattern* [12].

The relational operators can be used to compare values of Pascal enumerated types, and the predefined functions `ord(X)`, `pred(X)`, and `succ(X)` return the ordinal number, predecessor, and successor of `X`, respectively. A `for` statement can iterate over the values of an enumeration (or a subrange of them), and enumeration constants can be used as selectors in a `case` statement and to index arrays.

An important feature of Pascal's enumerated types is that they are *typesafe*: each enumerated type is different from and incompatible with all other types, including integers and other enumerated types. The compiler can enforce strong typing by ensuring that only the enumerated values of an enumerated type can be assigned to a variable of that type or passed as a parameter to a function or procedure that is defined to accept an argument of that enumerated type. Pascal's enumerated types are also *secure*: given the declaration of type `Day` above, `succ(Sun)` and `pred(Mon)` are compile-time errors.

Subsequent languages introduced variations of the enumerated type. Several languages, including C++ and Ada, allow the ordinal values of enumeration constants to be specified, as in

```
enum coins {penny = 1, nickel = 5,
            dime = 10, quarter = 25};
```

Ada and C# allow enumerations to be overloaded (i.e., the same literal can be used in the declaration of more than one enumeration in the same scope); C# requires that overloaded names be fully qualified, but Ada allows the type prefix to be omitted if the compiler can infer it from the context [15].

Java did not include support for an enumeration type until release 1.5 (Java 5.0, 2004). Prior to its introduction, many Java programmers used the ad hoc solution that is referred to as the *`int` enum pattern* [12]:

```
public static final int MON = 0;
public static final int TUE = 1;
  . . .
public static final int SUN = 6;
```

Several writers suggested ways of implementing typesafe enumerations using Java classes [16]–[20]. In its "Taxonomy of Problems in Teaching Java" [21], the ACM Java Task Force identified the lack of enumerated types as a "weakness in the Java type system."

The `enum` type introduced in Java 5.0 shifted the concept of an enumerated type away from a set of integer constants toward the concept of classes that is central to object-oriented programming [22, p. 112]. A Java `enum` type is a class of which "the possible values of the enumeration are the only possible instances of the class" [23, p. 257]. All `enum` types are implicitly subclasses of the predefined class `Enum`, from which they inherit `toString`, `equals`, `compareTo`, `hashCode`, `ordinal` (which returns the zero-based position of the constant in its `enum` declaration), and a few other methods. In addition, for each `enum` type, the static methods `valueOf(String)` and `values()` are implicitly declared; the former returns the enumeration constant corresponding to a name, and the latter returns all the constants of the enumerated type [24, §8.9].

Java enumerated types can have constructors, instance data fields, and methods. The combination of a constructor, an instance field and an accessor can be used to associate values other than the default ordinals with enumeration constants, as illustrated in this example from the *Java Language Specification*, 3e [24, §8.9]:

```
public enum Coin {
  PENNY(1), NICKEL(5), DIME(10),
    QUARTER(25);
  Coin(int value) {this.value = value;}
  private final int value;
  public int value() {return value;}
}
```

An `enum` type may not be declared `abstract`, but it may include an abstract method if all the enumeration constants have class bodies that provide concrete implementations of the method. This feature can be used to implement an enumeration of arithmetic operations [24, §8.9] [12, p. 152], as shown in Figure 1. This simple example deals only with binary operators, but Java's enumerated type facility can easily be used to implement nullary, unary, and ternary operators as well as binary operators, and to distinguish between left- and right-associative operators and between prefix and postfix operators.

3. An Enumerated Type for Arithmetic Operators of Various Arities

3.1 A Simple Enumeration for Arities

The first step in supporting operators of various arities is to add an `Arity` enumerated type as a nested type within an `Operator` enumeration:

```
public enum Arity {
  NULLARY, UNARY, BINARY, TERNARY,
    GROUPING
 }
```

Implementing an `Operator` enumeration is like designing a *little language* [25]: we can decide what operations can be included in arithmetic expressions that use our `Operator` class to process them. The `Arity` enumeration above allows us to support unary operators such as factorial (`'!'`) and percent (`'%'`) and single-argument functions (such as `abs`, `sqrt`, `sqr`, `trunc`, `round`, `log`, `lg`, `ln`, `sin`, `cos`, and `exp` or `pow`); binary operators such as exponentiation (`'^'`) and modulus (`mod`) in addition to the usual '`+`', '`-`', '`*`', and '`/`'; and ternary operators such as C and Java's conditional operator '`?:`' and variations on that operator such as `pos`, `neg`, and `zero`.

The `NULLARY` arity is provided for pseudo-operators such as constants (`e` and `pi`) and no-argument functions such

```
public enum Operation {
  PLUS   { double apply(double x, double y) {return x + y;} },
  MINUS  { double apply(double x, double y) {return x - y;} },
  TIMES  { double apply(double x, double y) {return x * y;} },
  DIVIDE { double apply(double x, double y) {return x / y;} };

  abstract double apply(double x, double y);
}
```

Fig. 1: An enumeration with an abstract method.

as `rand`. Strictly speaking, constants such as *e* and *pi* are operands, not operators, but in the context of an expression evaluator, they could be considered as no-argument functions that return a result, so I include them as “operators” for the convenience it provides for writing a parser for arithmetic expressions: all non-numeric tokens can be processed by calling the `eval` function on the `Operator` represented by that token.

The `GROUPING` arity constant was added to designate symbols such as parentheses ‘`(`’ and ‘`)`’ and the ternary operator separation symbol ‘`:`’. It is an error to attempt to evaluate these symbols, and providing a separate arity category for them facilitates our handling of that error.

3.2 A Constructor for `Operator` Enumerations

Operators have other attributes in addition to arity, such as priority, associativity, and location (prefix, postfix, or infix).

Associativity and location can be supported with additional nested enumerated types:

```
public enum Associativity {
  LEFT, RIGHT, NA
}
public enum Location {
  PREFIX, INFIX, POSTFIX,
    PRE_OR_POSTFIX, NA
}
```

Neither associativity nor location are applicable to nullary pseudo-operators or to grouping constructs[2]. Some operators (such as ‘++’ and ‘– –’) can appear in either prefix or postfix locations.

With the definitions of these enumerations and some attributes, the constructor for the `Operator` enumerated type could be implemented as shown in Figure 2.

3.3 Eliminating Case Logic

One way to handle operators of differing arities would be to define multiple `eval` functions, each accepting a different number of arguments, as illustrated in Figure 3.

The body of each `eval` function would include a switch on the value of the enumeration constant, such as:

```
switch (this) {
  case PERCENT:
     return arg / 100.0;
  case ABSOLUTE_VALUE:
     return Math.abs(arg);
  ...
}
```

Each `eval` function should begin with a test of the arity of the enumeration constant on which the `eval` function was called, to ensure that the proper number of arguments have been passed, and throwing an exception if not (perhaps a customized `OperatorException`).

While this approach is workable, it is contrary to one of the fundamental ideas of object-oriented programming: the replacement of case logic by polymorphism. Instead of using conditional logic based on the type of an object, we use subclasses (or, as in the case of our `Operator` enumeration, different enumeration constants) to create objects whose behavior varies based on their type.

A common way to implement polymorphic behavior is to define an abstract method in the base class of a type hierarchy (or in the enumeration type) so that each subclass (or each enumeration constant) has to provide a concrete implementation of that method. In the case of our `Operator` enumeration that includes operators of various arities, the problem is that we don’t want to have to declare four different abstract `eval` methods, one for each possible arity, because then each enumeration constant will have to provide concrete implementations of all four (three of which will throw an exception to indicate that an incorrect number of arguments was provided, given the arity of the operator). How do we declare a single abstract method for the top-level enumerated type when different enumeration constants require different numbers of arguments?

Two solutions to this problem come to mind. The first uses another feature that was added to Java in release 1.5: support for *variable arity methods*, or `varargs` [12, p. 197–200]. By combining this feature with an abstract `eval` method in a Java enumeration, it is quite easy to support the evaluation of operators of various arities in an object-oriented way.

The second solution is to pass a reference to a collection

[2] Although the ‘:’ of the ternary operators is located between the second and third operands, and is thus strictly speaking an infix symbol, it is not a true operator, but a separator provided for the convenience of the parser, so the concept of location is not applicable in the same sense as it is for operators that can be evaluated.

```
private final String symbol;
private final int precedence;
private final Arity arity;
private final Associativity associativity;
private final Location location;

private Operator(String sym, int prec, Arity ar, Associativity assoc, Location loc)
{
  symbol = sym;
  precedence = prec;
  arity = ar;
  associativity = assoc;
  location = loc;
}
```

Fig. 2: The constructor for an `Operator` enumeration.

```
public double eval() { ... }
public double eval(double arg) { ... }
public double eval(double lhs, double rhs) { ... }
public double eval(double test, double ifTrue, double ifFalse) { ... }
```

Fig. 3: Multiple `eval` methods for operators of differing arities.

that contains the arguments to which the operator is to be applied, and to make it a responsibility of the `eval` method to retrieve the correct number of arguments from the collection (typically a stack).

3.3.1 An `eval` function using `varargs`

Using Java's `varargs` facility, the case logic of the example above can be avoided by providing a constant-specific implementation of the `eval` function in the body of each enumeration constant.

First, an abstract `eval` method is specified in the body of the `enum Operator`:

```
abstract double eval(double... args)
  throws OperatorException;
```

Because the method accepts a variable number of arguments, a single abstract `eval` method suffices for operators of differing arities.

Then each enumeration constant must implement the `eval` function, checking that it has been passed the correct number of arguments. A nullary `Operator` enumeration constant might be implemented as shown in Figure 4.

The implementation of a unary operator, such as `SQUARE` (represented by the symbol `sqr`), would be similar, but would check that `args.length` is 1 and would return a result such as `args[0] * args[0]`. A binary operator such as `MULTIPLY` would return `args[0] * args[1]`, and a ternary operator such as `NEGATIVE` (represented by the symbol `neg` and the separator ':') would return the value of

```
(args[0] < 0.0) ? args[1] : args[2].
```

We could make implementation of the `eval` method in each enumeration constant more concise by delegating to the `Arity` enumerated type the task of checking that the number of arguments passed matches the arity of the operator. To do this, we add the following attributes and methods to the body of the `Arity` enumeration:

```
private final int numArgsExpected;
private Arity(int numArgs) {
  numArgsExpected = numArgs;
}
abstract void checkArgs(int numArgs,
    String sym)
  throws OperatorException;
```

Each `Arity` constant then implements the abstract method `checkArgs`. The test that the number of arguments supplied matches the arity of the constant is the same for each arity, but the message passed to the `OperatorException` constructor is customized for each enumeration constant. For example, Figure 5 shows the implementation of the `checkArgs` method for the `UNARY` constant of the `Arity` enumeration.

The implementation of the `eval` function in each enumeration constant can then be simplified by replacing the `if` statement that compares `args.length` to the expected number of arguments with the following method call:

```
arity().checkArgs(args.length,
    toString());
```

3.3.2 Passing an Operand Stack to `eval`

A postfix expression evaluator or an infix expression parser typically uses a stack to keep track of the operands

```
RANDOM("rand", 7, Arity.NULLARY, Associativity.NA, Location.NA) {
  @Override
  double eval(double... args) throws OperatorException {
    if (args.length != 0) {
      throw new OperatorException("Pseudo-operator '" + symbol
        + "' accepts no arguments.");
    }
    return Math.random();
  }
}
```

Fig. 4: Implementation of a nullary `Operator` enumeration constant.

```
UNARY(1) {
  @Override
  void checkArgs(int numArgsProvided, String symbol)
      throws OperatorException {
    if (numArgsProvided != arityValue()) {
      throw new OperatorException("Operator '"
        + symbol + "' requires a single operand.");
    }
  }
}
```

Fig. 5: Implementation of the `checkArgs` method in an `Arity` constant.

that have been seen but not yet passed to an operator or that have been produced by evaluation of a subexpression. Instead of passing individual operands as parameters of the `eval` method, we could pass the operand stack[3] as the sole argument of `eval`.

To implement this method of passing operands to an `Operator` enumeration constant, we declare `eval` as an abstract method of the `Operator` enumeration, accepting a stack[4] argument:

```
abstract double eval
    (Deque<Double> valStack)
        throws OperatorException;
```

A concrete implementation of the `eval` method must then be provided in the constructor of each constant of the `Operator` enumeration. Figure 6 shows how `eval` might be implemented for the unary `round` operator, represented by the `ROUND` enumeration constant.

While the implementation of `eval` for a unary operator as shown in Figure 6 is quite simple, it is a little more complicated to pop the required number of arguments off the stack for a binary or a ternary operator because we also have to be careful to use them in the correct order: the first argument popped is the last one that was pushed. If we have more than a few binary or ternary `Operator` constants, we could factor out the repeated code by defining an abstract `getArgs` method in the `Arity` enumeration:

```
abstract double[] getArgs
    (Deque<Double> valStack);
```

We then have to marshall the correct number of arguments only once for each arity. Figure 7 shows a simple method of doing so, using the implementation of `getArgs` in the `BINARY` enumeration constant as an example. This implementation puts the arguments in the correct order in an array of `doubles`, the array being the same size as the arity.

Figure 8 shows how the use of `Arity.getArgs()` makes the code for the `eval` method of a binary `Operator` quite concise; the implementation of `eval` for a ternary operator is equally simple.

However, this simplicity is somewhat deceptive, because the implementation shown does not check that the stack actually contains at least as many operands as required by the operator, given its arity. If the user of the `Operator` class did not provide enough arguments for the evaluation of a specified operator, the `Deque.pop()` method will throw a `NoSuchElementException`; the client class should catch this exception and take the appropriate action, such as informing a user that the infix expression they entered contained a syntax error (namely, insufficient arguments).

[3] A postfix expression evaluator typically uses a single stack for both operands and operators, while an infix expression evaluator typically uses separate stacks for operands and operators.

[4] The method parameter is declared using the `Deque` interface because the Java `Stack` class has been deprecated; instead, use of the `Deque` methods that conform to the stack protocol for accessing a deque (`push, pop, peek`) is recommended. The concrete class `ArrayDeque` implements a deque using a circular array.

```
ROUND("round", 6, Arity.UNARY, Associativity.RIGHT, Location.PREFIX) {
  @Override
  double eval(Deque<Double> valStack) throws OperatorException {
    return Math.rint(valStack.pop());
  }
}
```

Fig. 6: Implementation of `eval` for a unary operator.

```
BINARY(2) {
  @Override
  double[] getArgs(Deque<Double> valStack) {
    double[] result = new double[2];
    result[1] = valStack.pop();
    result[0] = valStack.pop();
    return result;
  }
}
```

Fig. 7: Implementation of `getArgs` for the `BINARY` arity.

```
DIVIDE("/", 5, Arity.BINARY, Associativity.LEFT, Location.INFIX) {
  @Override
  double eval(Deque<Double> valStack) throws OperatorException {
    double[] args = arity().getArgs(valStack);
    return args[0] / args[1];
  }
}
```

Fig. 8: Implementation of `eval` for a binary operator.

4. Teaching Arithmetic Expression Evaluation

It would probably be unreasonable to expect a first-year computing science student to implement an `Operator` enumerated type as described above. However, I have successfully used an `Operator` enumeration in an assignment on infix expression evaluation in an introductory data structures course by providing the source code for `enum Operator` to the students and showing them how to use it to apply operators to operands. Their assignment was to implement an infix expression parser and evaluator that correctly handles operators of various arities and associativities. I provided the students with pseudocode for an algorithm to evaluate infix expressions that use only left-associative binary operators; the challenge of the assignment was to implement the algorithm in Java and to revise it to correctly handle right-associative operators such as exponentiation (‘^’) and unary and ternary operators. Such revisions are fairly easily made when some thought is given to the problem, and students usually have to think through the problem because they may have difficulty finding sample code on the internet for arithmetic expression evaluators that handle right associativity and multiple arities.

It is valuable for students to write (or revise) an infix expression evaluator because it helps them to understand how expressions that they write in their programs will be evaluated. For example, when asked about the order in which an expression will be evaluated, students will commonly say that parenthesized subexpressions will be evaluated first. Writing or studying an infix expression evaluator will quickly show that this is a misconception. For example, given the expression $2*3+(4-5)$, a typical infix expression evaluator will evaluate $2*3$ first, pushing the result to an operand stack.

This assignment also acquainted the students with enumerated types in general and with the enhanced features of the Java enumerated type facility through studying the `enum` code that was provided and learning to call the public methods of the `enum` such as `isUnary()`, `isLeftParend()`, `precedence()`, `isRightAssociative()`, and `eval()`.

Such an example would also be useful in a programming languages course to illustrate the difference between traditional `enum` facilities in languages such as Pascal, Ada, and C++ and the more object-oriented approach taken by the designers of the enumerated type facility introduced in Java 5.0. This also provides a good opportunity to contrast the use of case logic with dynamic dispatch and polymorphism.

Several variations on this assignment might also provide good learning opportunities. For example, an infix expression evaluator could also be designed as the combination of two other applications: an infix-to-postfix converter and a postfix expression evaluator. This is the approach taken in Kruse and Ryba's [11] and Koffman and Wolfgang's [6] case studies.

Goodrich and Tamassia [8, Exercise C-7.15] suggest that infix-to-postfix conversion can be done by converting an infix expression to its equivalent binary tree representation using an algorithm that they provide to construct an expression tree from a fully parenthesized arithmetic expression. The equivalent postfix expression can then be generated by a postorder traversal of the expression tree. Lambert and Osborne [3] present this method in greater detail. Preiss [2] takes the opposite approach, constructing an expression tree from a postfix expression, then emitting the corresponding infix expression. Drozdek [7] provides pseudocode for generating assembly code or intermediate code in a compiler from an expression tree; such an exercise might be relevant in a programming languages course as well as a data structures course. Expression trees are typically discussed in connection with binary trees, so the arithmetic expressions featured in such examples are restricted to using binary operators.

Students could also be asked to add operators to the `Operator` enumeration; this would require them to think about the relative priority and the associativity of the operators to be implemented, in order to ensure that the order of execution of expressions including the new operators corresponds to the expected or specified behavior of the operators.

Kruse and Ryba [11] also discuss the *running-sum condition* for properly formed postfix expressions; students could be asked to implement this check when parsing postfix expressions.

5. Conclusions

Java enumerated types can be used to implement a fully featured object-oriented `Operator` class that facilitates the use of operators of varying arities, precedences, associativity, and locations of use.

It is possible to specify a single abstract `eval` method in such an enumeration that is concretely implemented in enumeration constants that represent operators of differing arities, rather than using case logic, either by using the `varargs` facility or by passing an operand stack as an argument to the `eval` function and having each operator (or the `Arity` member of each `Operator` enumeration constant) pop the appropriate number of arguments off the stack.

The use of such a class in a data structures course exposes the students to enumerated types while allowing them to focus primarily on the algorithms for parsing and evaluating arithmetic expressions using a stack. Such an example could also be used in an algorithms course to introduce infix-to-postfix conversion or postfix expression evaluation, or in a programming languages course to contrast the object-oriented nature of Java's enumerated type with the `enum` facilities of other languages or to illustrate the generation of code for expression evaluation.

References

[1] D. A. Buell, *Data Structures Using Java™*. Burlington, MA: Jones & Bartlett Learning, 2013.

[2] B. R. Preiss, *Data Structures and Algorithms with Object-Oriented Design Patterns in Java™*. New York: Wiley, 2000.

[3] K. A. Lambert and M. Osborne, *Java™: A Framework for Program Design and Data Structures*. Pacific Grove, CA: Brooks/Cole, 2000.

[4] N. Dale, D. T. Joyce, and C. Weems, *Object-Oriented Data Structures Using Java*, 2nd ed. Sudbury, MA: Jones & Bartlett, 2006.

[5] P. Smith, *Applied Data Structures with C++*. Sudbury, MA: Jones & Bartlett, 2004.

[6] E. B. Koffman and P. A. Wolfgang, *Objects, Abstraction, Data Structures and Design Using Java™*. New York: Wiley, 2005.

[7] A. Drozdek, *Data Structures and Algorithms in Java™*. Pacific Grove, CA: Brooks/Cole, 2001.

[8] M. T. Goodrich and R. Tamassia, *Data Structures and Algorithms in Java™*, 5th ed. New York: Wiley, 2005.

[9] R. Sedgewick and M. Schidlowsky, *Algorithms in Java, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*, 3rd ed. Boston: Addison-Wesley, 2003.

[10] M. A. Weiss, *Data Structures and Problem Solving Using C++*, 2nd ed. Reading, MA: Addison-Wesley, 1999.

[11] R. L. Kruse and A. J. Ryba, *Data Structures and Program Design in C++*. Upper Saddle River, NJ: Prentice Hall, 1999.

[12] J. Bloch, *Effective Java™*, 2nd ed. Upper Saddle River, NJ: Addison-Wesley, 2008.

[13] K. Jensen and N. Wirth, "Enumerated and subrange types," in *Pascal User Manual and Report*. New York: Springer, 1991, pp. 50–54. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-4450-9_6

[14] B. J. MacLennan, *Principles of Programming Languages*, 3rd ed. New York: Oxford University Press, 1999.

[15] M. L. Scott, *Programming Language Pragmatics*, 3rd ed. Burlington, MA: Morgan Kaufmann, 2009.

[16] J. Bloch, *Effective Java™Programming Language Guide*. Boston: Addison-Wesley, 2001.

[17] P. A. Cairns, "Enumerated types in Java," *Software—Practice and Experience*, vol. 29, no. 3, pp. 291–297, 1999.

[18] E. Armstrong, "Create enumerated constants in Java," *JavaWorld*, Jul. 1997. [Online]. Available: http://www.javaworld.com/article/2076970

[19] P. Bishop, "Java tip 27: Typesafe constants in C++ and Java," *JavaWorld*, Mar. 1997. [Online]. Available: http://www.javaworld.com/article/2077524

[20] T. E. Davis, "Use constant types for safer and cleaner code," *JavaWorld*, Sep. 1999. [Online]. Available: http://www.javaworld.com/article/2076481

[21] ACM Java Task Force, "Taxonomy of problems in teaching Java," Feb. 2004, unpublished draft. [Online]. Available: http://www-cs-faculty.stanford.edu/~eroberts/java/java-problem-taxonomy

[22] A. B. Tucker and R. E. Noonan, *Programming Languages: Principles and Paradigms*, 2nd ed. New York: McGraw-Hill, 2007.

[23] R. W. Sebesta, *Concepts of Programming Languages*, 10th ed. Boston: Addison-Wesley, 2012.

[24] *The Java Language Specification*, 3rd ed., Sun Microsystems, Inc., 2005. [Online]. Available: http://docs.oracle.com/javase/specs/jls/se5.0/html/j3TOC.html

[25] J. Bentley, "Little languages," *Communications of the ACM*, vol. 29, no. 8, pp. 711–21, Aug. 1986.

Helping Students Succeed by Improving CS1 - A Case Study

Robert Hatch
The University of Virginia's College at Wise
One College Ave.
Wise, VA 24293
rjh7g@mcs.uvawise.edu

ABSTRACT

This paper discusses an attempt to boost retention and better understanding of programming concepts by making changes to a CS1 course. These changes included nightly homework, more exams, and the addition of a lab hour where students worked in teams to complete activities assigned. Prior offerings of the course assigned weekly programming assignments, three exams over the course of a semester, and a final exam. The changes made to the course are compared to prior offerings of the course, based on student performance, as well as student feedback on the new aspects of the course. The course continues to be tweaked, based on student feedback, and the instructor's in-class observations.

Keywords

C++, CS1, lab, student enhancement

1. INTRODUCTION

Within our department, there has been a shift in of focus to find ways to both recruit more information technology (IT) majors, as well as find ways to boost retention of students who enter as IT majors. Our department loses many of the students who take CS1 or CS2. Some of the reasons behind the loss of potential majors include their inability to understand how to program, which could be a result of misgivings in high school, prior to their arrival at college. Some students enter an IT field based on the reason that they play video games and think they would like to major in the field, without having any notion of all of the steps and involvement necessary to create a successful video game.

One potential solution was to implement a lab hour, in addition to the three hours per week the students meet for class. The lab hour perhaps needed to be a way to help students with concepts they were struggling with during lectures. The department hoped that this lab would help retain more students as well as provide them with a better foundation on programming principles before proceeding to our CS2 course.

The remainder of the paper is divided up as follows: Section 2 discusses related work; Section 3 discusses the previous format of our CS1 course, as well as our new format with the lab; Section 4 shows changes in student performance, as well as feedback about the new format; and Section 5 ponders future work.

2. RELATED WORK

The biggest factor considered was the addition of the lab to the CS1 course. For students to succeed, the lab, in combination with daily homework and quizzes, was critical.

2.1 Introduction of a Lab

Brown [2] believed that "most courses at the (CS1) level should include laboratory sessions. These laboratories provide the student with the opportunity to gain practical experience by actually solving problems on the computer." Kumar notes that the Curriculum 91 report recommended including hands-on lab experiences in introductory programming courses. The idea of closed labs comes from this report; these are fixed-duration lab sessions during which students work on assignments under the supervision of faculty, in addition to their class assignments.

McCauley, et al. [8] were curious whether the effectiveness of the lab hinged on the fact that faculty in charge of the lecture are also in charge of the lab. In terms of performance, both sets of students performed "equally well" when they had the same person as lecturer and lab instructor than when they had different faculty members in each position. Ford, et al. [3] consider that the lab gives the instructor an opportunity to see how well the lectured material is learned and to work one-to-one with students, producing major benefits: group work, collaboration, collaborative problem solving, and work in a controlled setting. Labs can be beneficial to students, if they are structured properly, because labs are more than just "homework assignment completion" and "teacher-assisted debugging sessions," as suggested by Beaubouef and Mason[1]. Patitsas and Wolfman [10] discuss how to create effective closed labs in early CS courses. The two note that CS educators believe labs might "dramatically" improve interest in and perception of CS but also can dissuade students if the lab isn't helpful.

LeJeune [7] suggests that collaborative learning is a valuable educational tool for use in CS1. Collaborative learning is defined as instructional methods whereby students are encouraged or required to work together on academic tasks.

2.1.1 Types of Labs to Consider

Soh, et al. [11] points out that labs foster cooperative learning, which is an "effective pedagogy for CS, producing significant gains in student achievement." The authors note that cooperative learning helps students understand that working in small groups helps them succeed in achieving tasks as well as develop communication and problem-solving skills.

Walker [12] provides a roadmap of how to conduct a lab that is based in active-learning experimentation. The observed benefits of the lab include improved student syntax; positive grades and positive, confident students; active learning; and more time for faculty to contact and get to know students.

Lasserre [6] discusses the benefits of using team-based learning (TBL) on a first-term programming class. The first experiences with computer science and programming should be positive for students; their experience further shapes their attitude toward the field and how it impacts the interest and success of the student. Team-based learning (TBL) is a teaching technique that places emphasis on applying concepts as a means to learn. Its main objective is to test the current understanding of concepts and applying newly-acquired concepts on increasingly difficult problems, but students need to come to class understanding concepts prior to the lecture. Instructors found that students understood concepts better when they discussed them with others in class.

Another direction to consider for instruction is the use of studio-based learning (SBL). Narayanan, et al. [9] defines SBL as an instructional technique that "emphasizes collaborative, design-oriented learning." SBL might have some merit in CS1 courses. In a separate paper, Hundhausen, et al. [4] explains the phases of the pedagogical code reviews (PCR) as an extension to SBL: write, submit, review, feedback, revise, and quality assurance. The PCR follows a similar path to a formal code inspection.

2.2 Classroom Changes

A lack of practice and feedback exists in classes and labs. Students have a right to have a knowledgeable programmer look at their code and provide meaningful feedback. Instructors also might give more tests and pop quizzes if instructional time isn't lost in class or on grading. Students have poor project management skills; they are unable to estimate the amount of time taken to complete a project, and they wait until the last minute to start on assigned work [1].

Problem solving skills and logical reasoning required to be successful in CS1 are mainly developed through large amounts of practice [6]. Additionally, in terms of practice programming, the greater number performed, the easier problems seemed when they were encountered later [5].

3. COURSE FORMAT

This section details current and previous formats of our CS1 course.

3.1 Prior Offerings of CS1

Our CS1 course was a four-hour class that met three hours a week. The course was taught using the following components comprising the course grade: programming assignments, exams at regular intervals, and a final exam, with a smaller portion of each student's grade based on participation and quizzes, depending on the instructor. Regular exams and programming assignments made up significant percentages of the student's final grade.

3.1.1 Grading

The programming assignments consisted of using tasks that the students had learned recently, based on when the assignment was given. The exams consisted of these types of questions: short-answer; correcting code questions; short programming questions; a tracing question; and a long programming question. The short-answer questions asked students about concepts and sometimes asked them to provide examples to further explain concepts. The correcting code questions tasked the students to present a piece of code (approximately five lines) and find the problems with the code, if any existed. If problems were present, students had to supply the correct code. The short coding questions would ask students to either write the necessary code snippets to answer the task specified or rewrite a given code snippet using a different structure (for instance, a "while" loop is presented; rewrite this "while" loop using a "for" loop). The tracing exercise required students to provide output for a given code. The students earned points if they showed their work. The last component, the long coding problem, asked students to write an entire C++ program to solve the given problem or task.

3.1.2 Class Structure

The structure of the class primarily used PowerPoint presentations and provided opportunities with in-class exercises for students so that they could apply their knowledge and understanding of concepts which had just been introduced. The order of topics coincided with the format of the textbook: variables and sequential statements, classes, selection and iteration statements, functions, arrays, pointers and references, and advanced class concepts.

Discussions of classes were split up because we wanted to give students a foundation for their use - that they were data types, just like integers and floating-point numbers, but were different because they contained multiple pieces of information. Classes could more adequately describe a "thing" than an integer, floating-point number, or character. Students had difficulty completing early programming assignments with classes because those concepts which would be covered several chapters later in the text were introduced in the chapter along with classes. This created the added frustration of learning several concepts at once.

File I/O was introduced as a means to give the students more interesting projects with large amounts of data. The student's job was to use compilation errors to figure out where and why the errors were occurring. Strings were introduced after arrays to give the students a sense that strings were just an array of characters. The discussion of strings led to giving students more interesting projects, similar to reasons why File I/O was introduced.

3.2 Fall 2012 Offering of the Course

The students' grades were assigned, based on a series of tasks completed during the course of a semester. These tasks included the aforementioned programming assignments, exams, and final exam, as well as a homework component, quiz component, and lab component.

3.2.1 Grade Components

The homework component consisted of assigning programming problems from the textbook that were due at a "random" time the first half of the semester, usually on a Monday with sufficient advanced warning. If students did not discuss or understand concepts presented in class, the homework due date was changed randomly to the end of class, and students had to finish and turn in their homework by the end of that class. This change was introduced mid-term. Grading homework was a quick "eye test," and students earned one of four possible grades: 100 - satisfactory completion; 50 - the program compiled but did not complete the task; 25 - the program did not compile; or 0 - no submission. The homework component counted as ten percent toward the final grade; the rationale was to make students practice more often and to avoid last-minute work on assignments.

Another addition was quizzes, given on or around the midpoint between exams; the first quiz was given at the end of the second week of class. Questions asked were often short-answer; the class had only half of a period (25 minutes) to complete it. Some questions were conceptual, while others required students to write code snippets. The purpose of each quiz was to serve as a diagnostic for what students needed to study for the next exam, so they only carried a weight of ten percent toward the final grade.

A student's lab grade was determined by his or her participation and impact in group discussions and work completed; the grades were assigned by other students in the group. The lab counted as 15 percent of a student's grade.

Exams were altered somewhat; the code-correcting section was dropped, and the long coding problem became a "practical problem." Thus the exam was split into two parts: a written component, which was closed-book, closed-note; and a practical component, which was open-book, open-note, open-computer and internet, but closed-neighbor. The practical component was graded as completed/works (full credit); compiles but doesn't completely do the necessary task (at least half-credit); and doesn't compile (normally one-third to one-fourth credit, unless there was a small syntactical issue). Four exams (excluding the final) were given over the course of the semester, based on the fact that students had to digest so much information during the semester. The exams were calculated as 30 percent of a student's grade.

The nature of the programming assignments remained the same, and so did the final exam. Programming assignments were weighted at 15 percent because they were the other previous primary indicator of whether a student would successfully complete the course. The final exam was worth 20 percent of the final grade.

3.2.2 Class Structure

The class was changed to de-emphasize the use of PowerPoint. Presentation files were available to students after class, but they only summarized points covered. The course made more use of writing points on a whiteboard, which encouraged students to take notes. The idea of having students complete in-class exercises and review examples of code remained. With the addition of quizzes, only half of available class time could be devoted to a concept. These class days became "problem days" when the instructor worked through at least one or two problems introduced at the end of each chapter in the textbook.

The lesson plan remains largely the same as prior course offerings; the sole difference is the introduction of object-oriented programming toward the end of the semester. The rationale for this change was based on the fact that students should be familiar and comfortable with many of the programming concepts covered, including writing and calling functions. Doing this allows the instructor to spend less time reviewing material previously covered other semesters related to the introduction of classes. This approach also allows the instructor to cover interesting object-oriented concepts in an unrushed setting, which is important for students to grasp when they move on to CS2. Previously, students might have received some instruction about destructors and default memberwise assignment. The way in which the course has been retooled allows students to receive adequate instruction in these concepts, as well as inheritance and composition.

Our department changed the course with the introduction of a lab component; as a result, students must meet for an additional hour. The initial intent of the lab was to have students work in groups to examine each other's code (code review) to expose them to other ways of thinking and understanding concepts. They were given checklists of questions each week, and each statement needed only a yes/no response. Checklist length increased as the semester continued; the students covered more and sometimes complex topics. We wanted them to see how others approached solving a problem. Some lab sessions ended with nominating a "best of group" code so that all students in the lab could see a particular solution and why it was voted "best of group." Sometimes the voting was the result of well-documented code; at other times, it was a unique solution. The last programming assignment became a two-week project, so the final labs involved working in groups to write software that addressed a problem assigned at the beginning of the lab period.

3.3 Spring 2013 Offering of CS1 Course

The grading scheme of CS1 did not change going into Spring 2013; all components from Fall 2012 were kept. The lab hour alternated every week between the aforementioned code reviews and a team coding exercise. The students responded more favorably when asked at the end of the Fall 2012 term about their experiences with the team coding activities but not the code reviews.

The homework component was changed; homework was still assigned at the end of every class, but it was due at the beginning of the next class (unless it was a scheduled exam day). Homework grading was altered. Randomly-examined homework was graded using the 100/50/25/0 structure, as discussed earlier. The remaining homework was graded for completeness, verifying that students had turned in the assignment. Students did not like the fact that homework was randomly due on a certain date without warning.

4. RESULTS

This section will detail the feedback received from students concerning alterations to the course, as well as suggestions for improvements. This feedback was obtained through a survey administered at the end of the course and was a mixture of quantitative feedback using a Likert scale, as well as qualitative feedback through open-ended questions. Additionally, student grades from previous semesters will be used in comparison to the students in the 2012-2013 school year to determine effectiveness of understanding.

4.1 Feedback from Students - Fall 2012

The quantitative feedback was elicited through twelve statements. A Likert scale accompanied each question, using a scale of 1-5, with a 1 meaning "Strongly Disagree" and a 5 meaning "Strongly Agree." Twenty-one students completed the survey. Table 1 represents the list of questions asked. The students were asked five open-ended questions. The first of these questions was, "If I could change one thing about the lab, what would it be? Why?" Responses included: having more lab activities where students were divided into teams to come up with a solution to a given problem; and having the teams complete the program by the end of the lab, then talking about unique solutions teams came up with during the next class period. Specific suggestions were made to change the code review to make everything electronic and to make the review items more specific than the others' "randomly" circling responses.

Table 1: Fall 2012 Quantitative Questions and Response Averages

1	The lab was a valuable component to the course.	3.62
2	The code reviews in the lab were helpful to see other students' solutions and ways of thinking.	3.43
3	I found working with others in the lab a valuable learning experience.	4.1
4	I was given adequate time to complete all of my teammates' code reviews.	4.43
5	I liked that each student was held accountable for the amount of effort put into the code reviews.	4.00
6	The labs reinforced concepts introduced during class time.	3.48
7	The homework helped me to understand concepts introduced in class.	3.76
8	I found the homework to be busy work.	3.38
9	The quizzes given were valuable to tell me where I stood, in terms of understanding concepts covered in class.	3.9
10	I would use the quizzes to study for the exams.	4.48
11	I found the workload for this course to be too much.	3.12
12	I liked that the exams were part written and part practical.	3.48

The second question posed to the students asked, "A couple of times during the course of the semester we had lab activities where teams worked together to work on a problem, instead of code reviews. Which do you prefer in the lab: the activities or the code reviews? Why?" The respondents sided with teamwork coding activities by a tally of 17-3 (two students liked both). The reasons for the team coding activities included students learn more with these activities, because they mimic the real world by working with others to solve a problem, plus everyone takes part in the project. Students who were in favor of code reviews felt like they received immediate feedback on what they had done, as well as ideas on other ways to solve a problem.

The third open-ended question was, "If I could change one thing about the homework assignments, what would it be? Why?" Some students were in favor of no changes; others were split over the randomness of when homework was due. Those in favor of the randomness liked being kept on their toes; those against it felt more pressure to get everything done.

The fourth question asked was, "If I could change one thing about the quizzes, what would it be? Why?" Eleven students stated the quizzes were fine "as is"; students liked the fact that the "quizzes (adequately) prepare you for exams." Others suggested the questions not be weighted so heavily.

The last question asked was, "If I could change one thing about the exams, what would it be? Why?" The most popular response to this question was that the students needed more time; two students suggested moving the exam time to a lab time. Two students suggested removing the tracing problem.

4.1.1 Discussion

Using the administered questionnaire and its results, some of the suggestions made by the students were implemented for Spring 2013. The first of these changes included expanding the lab activities to alternate weeks between code reviews and team coding activities. The students did seem more engaged with the coding activities, based on instructor observations in the lab. The questionnaire confirmed the students' enthusiasm in the lab with working together to solve a problem.

The other major change instituted for the Spring 2013 term was that homework would be assigned at the end of every class, as before, except that the homework was due at the beginning of the next class period. This change, hopefully, will encourage students to stay on their toes with respect to practicing and understanding concepts learned during the lecture. The only exception to the due date is when the next class period falls on an exam day; then the homework is due on the next lecture day that follows.

4.2 Student Feedback - Spring 2013 Semester

Quantitative and qualitative feedback again was elicited from students at the end of the semester. The previous semester's questionnaire was reused, with two additional quantitative questions that were included, to gauge students' thoughts on the coding activities added for this semester. Thirteen students completed the survey; the results can be found in

Table 2: Spring 2013 Quantitative Questions and Response Averages

1	The lab was a valuable component to the course.	3.62
2	The code reviews in the lab were helpful to see other students' solutions and ways of thinking.	2.69
3	I found working with others in the lab a valuable learning experience.	3.85
4	I was given adequate time to complete all of my teammates' code reviews.	4.62
5	I found the coding activities in the lab challenging.	4.08
6	I found working in groups on coding activities helpful to learning course material.	3.77
7	I liked that each student was held accountable for the amount of effort put into the team activities.	4.31
8	The labs reinforced concepts introduced during class time.	3.85
9	The homework helped me to understand concepts introduced in class.	4.08
10	I found the homework to be busy work.	2.69
11	The quizzes given were valuable to tell me where I stood, in terms of understanding concepts covered in class.	4.23
12	I would use the quizzes to study for exams.	4.69
13	I found the workload for this course to be too much.	2.85
14	I liked the fact that the exams were part written and part practical.	2.85

Table 2. As with the previous semester, the students agreed with the idea that they had adequate time to complete their code reviews; liked the fact that each student was held accountable for the amount of effort put into each lab activity; the homework was a helpful aid; and they studied for exams with the quizzes given in class. The students also agreed, albeit not as strongly, that the lab is valuable to the course; working with others is a valuable learning experience; and the labs reinforced concepts from class. The students also reaffirmed that the workload was not too much for them, just like their Fall 2012 counterparts. Because we added the group coding activities to the lab, Questions 5 and 6 were new. Students found the coding activities challenging, yet those exercises were helpful in learning course material.

The same five qualitative questions were posed to students that were asked in the Fall 2012 semester. A variety of responses addressed changing the lab. For example: one positive comment said, “In general, it allowed for more concepts to be openly discussed instead of reviewing the same general project 3 or 4 times.” Another student appreciated having more group programs than code reviews. A third student said that “the code reviews 'wasted' time that we could have used to practice more problems.” Generally, more group programming activities were suggested because code reviews were not helpful.

With respect to the second question (code reviews v. code activities), only two students favored code reviews; a third student found both types of activities helpful. Reasons in favor of code reviews included seeing how others came up with their solutions, hearing suggestions on one's own work, or just seeing other solutions in general. Lab activities were preferred for different reasons: they allowed general involvement from everyone in the group; gave a further understanding of concepts from class; enabled them to work as a team to solve a problem; if a student wasn't learning something new about a section, it re-enforced current knowledge; and helped students feel like they were learning something.

The third question posed to the students concerned changing the homework. Six students responded that they would not change anything. Four responded that they would like less homework assigned; one said that “homework on top of the programming assignments is far too much.” While one of these students was concerned about the volume, he or she did state that the homework did help with understanding the material better. One student suggested not having homework due on the same day as a programming assignment. Remaining issues included having the professor explain the homework beforehand because what the book stated was unclear, as well as a dislike for being graded on whether the homework submitted worked or not and a desire for more feedback about what went wrong.

Responses to the fourth and fifth questions remained consistent with student responses from Fall 2012.

4.2.1 Discussion

As with the administered questionnaire in the Fall, students remain closer to agreeing than being neutral about benefits of the lab. As with the responses to the Fall 2012 students, these students responded more positively toward group coding activities over code reviews. The lab component is still in its infancy, so using more group coding activities in Fall 2013 is probable, with even fewer code reviews. Code reviews, however, should not be totally shelved. It is important to see how others arrive at their solutions and emphasize there is more than one way to get from start to finish.

The slight attitude change for homework is encouraging. As evidenced from the Fall 2012 semester, students prefer structure over randomness concerning when homework assignments are due. The purpose of the homework was mainly to get the students to write code more frequently, beyond their programming assignments, with the hope that this increase in coding would result in a better understanding of material.

4.3 Comparing School Years 10-11 and 12-13

Table 3 shows the final letter grade students earned over the course of the semester. This table includes students who completed the course; it omits students with excessive absences or those students who stopped showing up closer to the end of the semester. The data suggest there might be benefits from revamping the course because students appear to be making higher grades. Student performance is shown in Table 4. The figures show significant improvement in exam averages and final exam averages with changes to the course. Since changes have been made to exams and finals themselves, numbers in Table 5 list correct percentage for each of the common areas of the exams (short answer, short coding, tracing, and long coding). As Table 5 notes, there

Table 3: Final Grade Distribution - 10-11 and 12-13

Semester	Students	As	Bs	Cs	Ds	Fs
Fall 2010	25	3	3	11	3	5
Spring 2011	19	1	3	3	5	7
Fall 2012	24	4	4	6	5	5
Spring 2013	17	5	1	5	4	2

Table 4: Performance - Exams and Assignments

Semester	Exams	Assignments	Final Exam
Fall 2010	57.19	72.83	44.04
Spring 2011	51.25	68.14	44.32
Fall 2012	68.66	77.38	68.02
Spring 2013	68.76	70.65	64.68

are some improvements seen in performance on the common areas of the exams. Tracing remains an issue to students. The long coding segment is also lower, in spite of the fact that this portion of the exam is open-book, open-note. Part of the problem with these two areas may lie in the fact that students feel rushed for time when getting to these last two sections.

5. CONCLUSIONS AND FUTURE WORK

Our department introduced the use of a lab to accompany our CS1 course as well as other changes to help students with the course material. The lab consisted of students reviewing one another's code in groups, as well as group coding activities. The students critiqued each other's impact on helpfulness and their active participation in a group setting. Other changes included daily homework assignments, which were collected the following class period. Quizzes between exams were added as a diagnostic to show a student's comprehension of topics. The exams also were altered to allow students an opportunity to complete a traditional "long coding problem" in the form of a practical question.

The data on final grades, as well as exams, show some improvement related to grades earned. Students still feel rushed for time when they must complete both a written portion and the practical portion of exams. This rush for time may contribute to poor scores in the tracing and long coding components of exams. Perhaps some homework could be geared toward more frequent tracing practice between exams as well.

Another year of study is required to see what further changes might be needed to maximize help provided to students so that they can understand the material. The idea of moving exams to a lab period might be feasible, as long as all students can show up for the same lab block. In our lab, students seem to find working in groups beneficial to their understanding. They respond more positively to group coding as opposed to the code reviews completed in groups. More group coding exercises may be introduced in future course offerings, but code reviews will not be completely eliminated from lab activities. The department feels that it remains important for students to see other ways of thinking and understand there often is more than one way to reach a solution.

Table 5: Exam Sections - Performance

Exams	Short Answer	Short Coding	Tracing	Long Coding
Fall 2010	54.43%	61.56%	44.76%	55.97%
Spring 2011	61.85%	64.02%	41.07%	49.01%
Fall 2012	71.79%	67.11%	50.3%	63.34%
Spring 2013	74.48%	70.44%	55.88%	64.6%

6. REFERENCES

[1] T. Beaubouef and J. Mason. Why the high attrition rate for computer science students: some thoughts and observations. *ACM SIGCSE Bulletin*, 37(2):103–106, June 2005.

[2] D. Brown. Cs 1 labs: goals and expectations. *ACM SIGCSE Bulletin*, 24(4):29–34, December 1992.

[3] F. Ford, L. King, L. Kovijanic, A. Kumar, M. D. LeBlanc, and L. Wilkens. The effectiveness of structured labs in cs1. *Journal of Computing Sciences in Colleges*, 17(6):6–9, May 2002.

[4] C. Hundhausen, A. Agrawal, D. Fairbrother, and M. Trevisan. Integrating pedagogical code reviews into a cs 1 course: an empirical study. *ACM SIGCSE Bulletin*, 41(1):291–295, March 2009.

[5] P. Kinnunen and L. Malmi. Cs minors in a cs1 course. In *Proceedings of the Fourth International Workshop on Computing Education Research*, pages 79–90. ACM, September 2008.

[6] P. Lassere and C. Szostak. Effects of team-based learning on a cs1 course. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 133–137. ACM, June 2011.

[7] N. LeJeune. Critical components for successful collaborative learning in cs1. *Journal of Computing Sciences in Colleges*, 19(1):275–285, October 2003.

[8] R. McCauley, C. Starr, W. Pharr, R. Stalvey, and G. Pothering. Is cs1 better with the same lecture and lab instructor. *ACM SIGCSE Bulletin*, 38(2):54–60, June 2006.

[9] N. H. Narayanan, C. Hundhausen, D. Hendrix, and M. Crosby. Transforming the cs classroom with studio-based learning. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 165–166. ACM, March 2012.

[10] E. A. Patitsas and S. A. Wolfman. Effective closed labs in early cs courses: lessons from eight terms of action research. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 637–642. ACM, March 2012.

[11] L.-K. Soh, A. Samal, S. Person, G. Nugent, and J. Lang. Analyzing relationships between closed labs and course activities in cs1. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, pages 183–187. ACM, June 2005.

[12] G. N. Walker. Experimentation in the computer programming lab. *ACM SIGCSE Bulletin*, 36(4):69–72, December 2004.

Lessons Learned from an Introductory Java Course for Non-Technical Students

M. Scaiano, A. Javadtalab, and L. Peyton
School of Electrical Engineering and Computer Science (EECS), University of Ottawa, Ottawa, Ontario, Canada

Abstract— *In this paper, we present lessons learned from teaching an introductory Java course to a group of students from diverse fields of study not related to computer science or engineering. In particular, we were interested in evaluating how such students conceptualize programming and what the most effective ordering of topics and course structure would be for them. The lessons learned have been drawn from observations, discussions, and self-reflection, as well as a student survey and the actual achievements of the students in the course. Based on these results, this paper recommends ways to improve the teaching of programming to students from non-technical disciplines.*

Keywords: Objects First, Guided Problem Solving, Introductory Java, Object Orientation, Non-Technical Students

1. Introduction

In this paper, we present lessons learned from teaching an introductory Java course to a group of students from diverse fields of study not related to computer science or engineering. The students lacked relevant experience and did not share common fields of study which made teaching the topic challenging. We applied and evaluated some recommended strategies from current research in an attempt to improve upon the way the course had been delivered in the past.

In particular, we were interested in evaluating:

1) How successful was the course at instilling confidence in students.
2) What was the most effective way to conceptualize programming for students.
3) What was the most effective order to teach the programming topics.
4) What was most effective structure (lectures, labs, assignments, projects, etc.) for delivering the course.

All three of the researchers were involved in teaching the course. There were discussions, reflections, and observations made throughout the course as we paid close attention to how students progressed in their learning and which aspects of the course were more or less successful in supporting that process. The students also filled out a survey at the end of the course that helped us evaluate the different techniques and approaches to teaching programming in the course. Based on the results of that process of self-evaluation, this paper recommends practical ways to improve the teaching of programming to students from non-technical disciplines.

The paper is organized as follows: Section 2 gives background on the problems faced in introductory courses and relevant research on how to address them. Section 3 gives an overview of the course and our evaluation methods. In addition, this section describes some of our teaching philosophies and how this was reflected in the course. Section 4 describes the results and our lessons learned: both philosophically and practically. Section 5 concludes the paper with a set of recommendations.

2. Background

In general, introduction to programming courses are known to be difficult for many students. This results in low retention rates and has prompted decades of ongoing research into alternative techniques for teaching programming [1] as well as research in the psychology of novice programmers and how they learn [2].

One difficulty students have with learning computer science is that they start without any mental models, unlike physical sciences where students have some idea how the world works (eg. physics: gravity and motion). Students tend to construct many inconsistent mental models before reaching a valid understanding of programming. The theory of constructivism claims “that knowledge is actively constructed by the student, not passively absorbed from textbooks and lectures” [3]. Thus teaching introductory computer science should focus on helping students build and validate their mental models.

Two large-scale studies highlight the difficulty that first year computer science students continue to have in learning how to perform relatively straightforward programming tasks. McCracken et al. [4] examined students’ abilities to program after they had completed one or two programming courses. 217 students from four universities participated in the test. The results were scored based on several criteria and they were aggregated into a general evaluation score. The average evaluation score for all tests was 22.9 out of 110.

Another study of 600 beginner students from seven countries showed that students were not skilled at tracing programs [5]. Participants were tested on twelve multiple choices questions. Two types of questions were used: the students were required to predict the result of a piece of code or to choose the correct piece of code to complete the question. The results were surprisingly poor, over a 25% of the students failed the most basic tests.

Perhaps the main reason for this failure is that the conversion of a concept to a set of instructions is not an easy task, especially for beginners. In fact, this essential skill for programmers has been characterized as a "wicked problem" [6].

Different approaches to this problem have been advocated. Some have advocated a focus on teaching computation independent of programming languages [7]. Others have advocated simpler, more natural languages, although it is not clear that this helps [8]. Four classic approaches are described in the ACM Computing Curricula 2001 (CC2001) report for teaching introductory programming. These approaches are now widely used in North America [9]. In [10], they describe three implementations of these approaches: imperative-first, functional-first, and objects-first. In their work they were in favor of the objects-first.

More recently, there has been a focus on problem-based learning [11], which allows beginners to learn programming in the context of solving well-structured problems from domains that they are familiar with and find appealing. In addition, different types of supporting material have been advocated such as web-based learning [12], distance learning facilities, and forum discussions to enhance the learning experience.

A recent movement in teaching has been the "flipped classroom" [13]. Where lectures and lessons are done at home, usually via a web based video or readings; the classroom is meant for problem solving and homework; the classroom is thus the opportunity for students to get assistance. While it is not clear that this is an effective approach for introductory computer science, it is being applied in a number of subjects and educational institutions.

3. Course Overview and Evaluation

The introductory course in Java programming was designed for first year undergraduate students not in the faculties of engineering or science. The 18 students came from different fields, such as English, GIS, Geography, Math, and History. For most students, this was their first exposure to programming (57%) (see Figure 1) and most lacked, or struggled with related topics, such as mathematics and logic. Only 2 students had actually taken any programming courses. This is of particular importance as the most influential factor in a students success in an introductory to computer science course is having previous programming experience [14].

The course was originally designed around a procedural (imperative) programming outline, including pseudo code and algorithm design. The class was taught by 3 researchers each proficient in Java and other languages. Each researcher believed in the importance of validating students' mental models and, as a result, each researcher used a highly interactive teaching style. Students were polled regularly about their expectations about code or their ideas about solutions to problems.

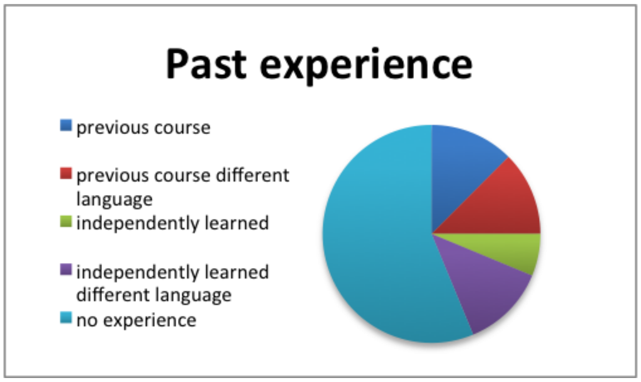


Fig. 1: Students' Past Programming Experience

There were two 2-hour sessions per week. One session was designated as a lecture and the other was designated as a lab. Each of the researchers took turns leading the lab, which allowed for flexible one-on-one assistance, as well as group lectures when an issue needed to be highlighted as students worked through their programming tasks related to the most recent lecture. For our University, this was a departure from the more common two lectures a week format.

We believed that programming should be taught with a significant practical component. Labs were opportunities to teach and encourage good programming practices, demonstrate common errors, and deal with practical issues. We make specific recommendations on this topic in section 5.2.

Three assignments, two tests, and a final project were given to the students; each is described in the following sub sections.

In addition to the four hours of official class time, each of the researchers had office hours where they met with students individually and in groups. Because of the small class-size, it was possible to meet with most students and get an accurate sense of their progress. The researchers met with students between 3-9 additional hours per week outside of class. The feedback from these meetings allowed for quick refinement of the teaching techniques. As the teaching techniques improved, the amount of assistance students required was reduced.

3.1 Test

The tests were given on the 7th week and 13th week of the course; approximately half away through and at the end of the course. Tests lasted 1 hour. Most questions required the students to manually trace programs and show the output. This effectively tested their ability to read and understand code, much like a reading comprehension test, and is a good method to validate mental models.

This is an important skill, because the better students are at reading code, the more easily they can find and understand errors. Tracing code and listing output style questions are easy to grade and do not require students to manually write perfect code. In addition this style of question allows students to get through longer tests, since they normally require hours of time with a compiler and editor to write simple programs.

3.2 Assignments

The first two assignments were a combination of planning algorithms in pseudo code and writing programs. Surprisingly, students did better at writing programs that they could test, than with writing pseudo code.

The third assignment was all programming which students performed very well on, except on one question where we significantly underestimated the difficulty. The evaluation of all assignments included a component dedicated to testing boundary conditions and program readability: extreme input testing, input near known conditions, good variables names, comments where needed, good use of functions (methods), correct indentation, and aligned braces.

3.3 Project

Students completed the project in groups of two. The project was a graphical version of the traditional board game “battleship”. The game was pre-implemented using an object-oriented model-view-controller architecture [1]. Sections of code were removed and students were given detailed instructions on what they should implement. The project spanned multiple files and involved numerous classes.

The instructions were organized such that as each function was implemented, it would enable new testable functionality in the user interface. Students were to implement each function in order and then test the effect. This provided helpful feedback to the students and would encourage them when they experienced the functionality they had implemented.

3.4 Course Outcome

The outcome of course was that all students felt they could somewhat read and write code, with more than 50% of students very confident in their abilities. The project encouraged students and improved their confidence. Students showed a higher aptitude for solving problems like the project than most of the “toy” problems in the assignments. The “toy” problems should have been easier, but they were abstract and disconnected from domains the students were familiar with.

Based our evaluations (assignments, tests, and project) most students succeeded in learning the course material and proved that they could read and write basic programs. This is best exemplified by the final project where all student groups, except one, earned grades greater than 70%; in fact, at least half the groups earned grades greater than 95%.

4. Lessons Learned

As the course came to an end, we requested students complete a survey to provide us with feedback. We had made a number of observations and hypothesises that we wanted validated or refuted. Some of the survey results were consistent with our expectations while others were surprising.

The major struggle for such students was building good mental models of programs and understanding how and why programs function or should function. Three teaching tools seemed to be particularly effective with our diverse group of students: guided problem solving, objects first, and teaching through labs and practical experience. The following three sub sections describe our three recommendations in detail and including our observations, student reactions/opinions, the effect of the approach.

4.1 Guided Problem Solving

Students struggled to understand and create pseudo code. This teaching tool which is suppose to abstract away the specifics of programming, did not seem well suited for these students. We believe that these students required something more concrete, testable, and less abstract. A number of survey results surprised us and helped us understand this issue.

All student indicated they could somewhat read and write code, but only 53% indicated they were confident that they read code, while 73% of students indicated they were confident they could write a program, if they knew the steps to take. This outcome surprised us, as students were more confident in their writing abilities than reading. There is recent research that explores the idea that code reading is a skill on its own and that one cannot just assume that it will be learned as a side of effect of learning how to write code [16]. This also highlights the difficulties students had in understanding code and constructing a mental model of a program. It should be noted that writing code is a constructive testable process, where reading code requires understanding each detail precisely and accurately without feedback.

When asked about the order of topics, 50% of students would have preferred harder topics earlier in the semester so that they would have more time to become comfortable with them. This is a surprising result since we would have expected all students to prefer to build up to difficult subjects. One possible factor is that more time is needed to grasp difficult topics and there is less time to do so if

[1] We started from an OpenSource battleship game [15] but redesigned much of the code to create a model-view-controller architecture. A practical benefit was that students could not use the original OpenSource code as a working solution.

one waits until later in the course to introduce the topic[2]. Another factor is that the difficult topics for students are precisely the ones needed to experience programming that is interesting and relevant to students (like the "battleship" project). Figure 2 shows the topics that students identified as the most difficult[3].

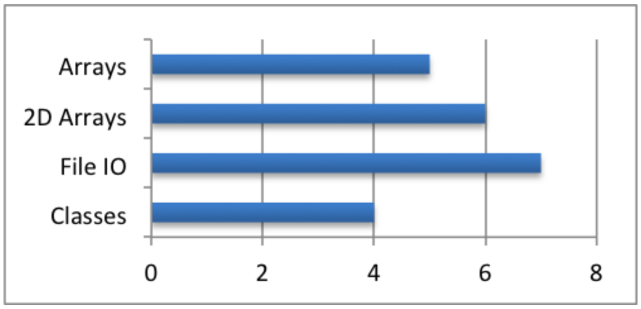


Fig. 2: The most difficult topics for students

When surveyed about their work load, as indicated in Figure 3, most students indicated they would have preferred more assignments with less work (63%). At the same time a significant number of students (33%) would have preferred (possibly alternatively) assignments with more coding but with detailed step-by-step instructions regarding the algorithms. This returns us to our original observation: students have trouble understanding the abstract concepts associated with programming and require more hands-on time to learn. We did not offer the option of having no projects, as the feedback on the project from students while they were doing it was overwhelmingly positive. Because of the work involved though, there was limited interest in having a second project.

These observations are in line with philosophies of constructivism as related to computer science. Students need to frequently test and validate their mental models. Furthermore, we do not believe students were suggesting less work (or to be lazy) instead they desired more focused opportunities to learn and practice programming. The use of more shorter focused assignments can be used to test, validate, and reinforce students understanding of specific concepts. Large assignments which test all aspects of a students understanding (syntax, branching, algorithm design) can lead to students getting stuck or confused without an ability to update their understanding or correct their errors.

From what we observed and heard from students, we would recommend a problem-based learning approach in which students are guided to learn concepts in the context of solving well-structured problems. Since students struggled to design and understand code, this encourages them to write more code, but with the algorithm (steps) clearly defined for them.

Assignments should be much like the project we assigned, the work should be clearly laid out and students can see and experience the outcome of their work. Instead of assigning "toy problems" for students, assignments can involve a comprehensive framework so that students can see a meaningful problem solved. Rather than writing all the code for a simple solution, students could instead fill in only the critical pieces of a significant piece of software that is already largely coded. Students exposed sooner to many difficult programming concepts are given more time get comfortable with them. Furthermore, this will build students' confidence in their abilities to read and write code.

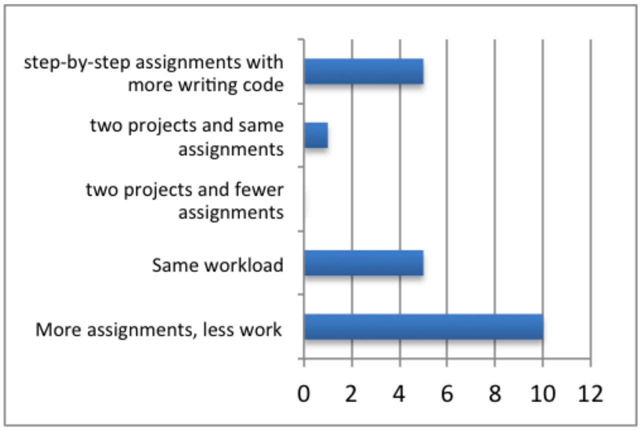


Fig. 3: Students' workload preferences

4.2 Objects First

Object orientation is the best and easiest way for students to solve interesting problems. As is well known, objects allow modular design because difficult concepts can be encapsulated and abstracted away for students. Furthermore, object orientation offers an intuitive structure that is easy for students to grasp. This is confirmed by other researchers findings and from the outcome of the class project.

Amongst the authors this was the most debated issue. Would the students be able to grasp the idea of an object? But as previously shown, students would have preferred harder subjects first, classes included. In addition 25% of students indicated in a partially form-free section of the survey that classes and objects should have been taught sooner.

We found students easily understood the idea of objects when taught and demonstrated with concrete examples. It seems in particular that object-orientation matched the way that non-technical students with limited mathematical and logic skills conceptualized how the computer program was functioning. It also helps abstract away from the low-level

[2] Another scheduling factor might be that students prefer the hard material earlier so they can fairly gauge their possibility of success before the final drop date.

[3] We decided not to cover recursion which is a difficult algorithmic concept for most students, especially first-year non-technical students taking an elective programming course.

language constructs particular to a language like Java that many students found difficult to grasp (e.g. File I/O). It can also help abstract away from complex structures, like arrays, so that students can work with complex programs earlier.

The project involved numerous classes and objects, something the students had had minimal exposure to, but students quickly grasped the idea and easily succeeded in the project. As noted earlier student groups earned excellent grades. While procedural programming can be taught using Java, the entire Java framework is object oriented and almost any use of the framework requires objects. Teaching objects demystifies a number of aspects of working with the Java framework and is particularly relevant when teaching harder topics such as file I/O, string manipulation, and arrays.

We recommend first teaching students how to use and interact with objects, while avoiding object-oriented design, design patterns, algorithmic thinking and topics related to planning a program. Objects are an intuitive structure that can model concrete ideas and be easily visualized, understood, and taught to non-technical students.

4.3 Labs vs Lectures

We recommend an introductory course structure focused on reading and writing code, instead of algorithm design. These skills should be primarily taught with hands-on experience instead of lectures. Our survey confirms what we believed, that labs and hands-on experience were the more helpful for students than lectures. Students were asked,“What were the three most helpful things to learning Java?” Their responses are shown in Figure 4.

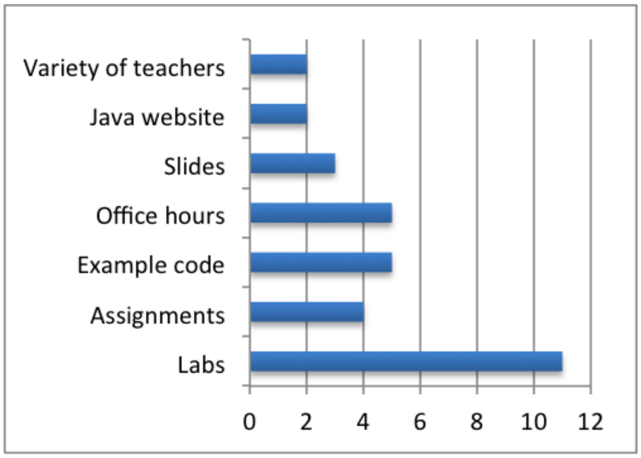


Fig. 4: Most helpful things for learning Java

Students found that above all else labs were the most helpful tool for learning and in a separate question that lectures were the least useful. We do not suggest doing away with lectures, nor did our students, instead we recommend keeping lectures closely paired with labs. Students clearly indicated that they preferred lectures followed by labs (on the same topic), though they disagreed on amount of time between lectures and labs: same session, a couple of days apart, or a full weekend.

Our observation was that if there were long delays between lectures and labs, students would not recall the theory taught in the lecture. Thus, our recommendation is to have lectures and labs the same day or even in the same session. A large portion of time spent with students should be on working through problems and validating their understanding.

4.3.1 Running Labs

We found that labs could be approached in three different ways: assign a problem to individuals or groups and assist as needed; assign a problem then have the class discuss a solution and afterwards allow them to implement it; assign a problem then guide the class to a solution and afterwards demonstrate the solution. Each approach was preferred by different segments of the class as indicated by the surveys.

The first approach worked well for students who quickly understood the lesson, problem, and solution. Most of students, could not figure out how to solve the problems, nor encode the solutions. Most of these labs had the researchers running from student to student repeating the same lesson.

The second approach, assigning a problem and having the class discuss the solution then having them implement it, reduced the work load compared to the first approach but many students could still not encode the solution in Java.

The third approach had the TA introduce a problem (sometimes the problem was framed within existing code). The TA would guide students to a solution and then demonstrate this solution on screen. Students would then repeat (re-type) what the TA had written. This helped students who could not figure out how to encode any aspect of the program, but also encouraged students not to try the problem for themselves. This approach frustrated students who wanted to resolve the problem independently. While one might assume this approach removes the need to assist students one-on-one, it is not true; students would often make small typos, invisible to their untrained eyes, that would cause compiler errors or runtime failure of the program.

Ultimately, the best approach was to start the labs with the third technique, this introduces the syntax and solutions. As the lab progresses through different examples, shifting to the second approach. Allowing students more time to work on the problems themselves before showing a general solution on screen.

We consider this to be a constructivist approach as we first demonstrate and model an idea then students validate their understand by working through the problems together. As students understanding (mental model) of the subject is refined, they are given more independence and larger tasks to work on.

This approach also allows an instructor to demonstrate

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

common errors and engage the class in a discussion of the cause and solution. Some previous studies have shown that this method can be useful for learning a new language [17] [18] [19].

5. Conclusions and Recommendations

In this paper, we presented our approach for teaching introduction to Java programming to students from non-technical fields. We have provided recommendations based on our successes and failures. Considering the learning outcomes and our survey results, students learned the basics of programming in Java.

Student response on the survey indicated 53% of students could confidently read Java, while the remaining 47% were somewhat able to. 73% of students said they would be able to write a program if they knew how to solve the problem. More than half the class (60%) was planning on taking another programming course, and another 26% were considering it. These results are particularly encouraging considering for most this was their first exposure to programming. Furthermore, with past programming experience being the most influential factor for success when taking programming courses, these students are now primed for future success.

5.1 Recommendation for Course Structure

Given the results of the survey and the students' performance over the semester, we recommend when teaching introduction to Java for students of such diverse fields in a 14-week course:

- Use guided problem-solving approach. Include 6 assignments with no pseudo code, just programming. Questions (particularly early in the semester) should clearly describe the steps required, thus focusing on students' understanding of the language, syntax, and structure and not on algorithmic thinking.
- Include a guided project with clear check points and visual or tangible effects. This will engage students and instill them with confidence. With such projects, students will rise to the occasion and be very successful.
- Teach objects first. This representation is intuitive, allows students to tackle large well structure problems, and is foundational to using the Java framework.
- Teaching should focus on reading and writing, not algorithm design. The primarily tool for teaching should be hands-on labs where students are able to write code and receive assistance.

5.2 Recommendation for Practical Labs

We used a number of simple but notable guidelines in our labs to encourage and assist students in maximizing their success. These are just practical programming practices that seemed to help our students.

Experienced developers and students alike tend to write entire programs or solutions then test them. While for experienced developers, this works effectively because they rarely make syntactic or logical errors and can always debug or unit test their code, this is not true of novices. Novice developers can make a dozen errors in 5 lines of code and without experience in debugging or reading code, they usually cannot resolve the errors on their own.

These pragmatic guidelines are designed to help students quickly identify and resolve errors in their programs. Instructors should demonstrate these guidelines because it will help students quickly identify flaws in their code and in their understanding of programming.

- Students should compile often (after one or two statements compile the code). This leads to quickly identifying syntax errors or use of undeclared variables.
- Don't just compile often, run often with "debug" or "trace" statements.
- Always test the simplest possible piece of code. Confirm even the simplest piece of code works; a misplaced semi-colon can make the right code wrong.
- Always put matching opening and closing braces. Students would often open braces but forget to close them. We encouraged them to immediately close a brace after opening one, then they could fill the body with statements.
- Always keep the correct indentation. As with braces, students did not have the trained eye to identify scope/context, thus encouraging and enforcing correct indentation in assignments helped them see scope.
- Encourage use of the online Java documentation [20] to look up method names and argument lists. This can build confidence and provide a practical experience of finding the information that they need.

References

[1] Baldwin, L.P.; Kuljis, J., "Visualisation techniques for learning and teaching programming", In *Proceedings of the 22nd International Conference on Information Technology Interfaces*, pp.83-90, 16 June 2000.

[2] Richard E. Mayer. *The Psychology of How Novices Learn Computer Programming*. ACM Comput. Surv. 13, 1 pp 121-141, March, 1981

[3] Mordechai Ben-ari, "Constructivism in Computer Science Education", in *Center for Research on Learning and Teaching*, University of Michigan, pp38-44, 1998

[4] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, and Tadeusz Wilusz. "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students". In *Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '01)*. ACM, New York, NY, USA, pp 125-180, 2001

[5] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik MostrŽm, Kate Sanders, Otto SeppŁlŁ, Beth Simon, and Lynda Thomas, "A multi-national study of reading and tracing skills in novice programmer", In *Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '04)*. ACM, New York, NY, USA, pp 119-150, 2004

[6] Degrace, P., & Stahl, L. H. *Wicked problems, righteous solutions: A catalog of modern engineering paradigms*. Prentice Hall. 1998

[7] T.J. Cortina, “An introduction to computer science for non-majors using principles of computation”, *ACM 38th SIGCSE technical symposium on Computer science education*, pp 218-222, New York, USA, 2007

[8] T. Wright, A. Cockburn, “Evaluation of two textual programming notations for children”, *AUIC 05, Sixth Australasian Conference on User Interfaces*, pp55-62, Australian Computer Society, Darlinghurst Australia, 2005

[9] Joint Task Force on Computing Curricula. “Computing Curricula 2001 Computer Science.” *Journal of Educational Resources in Computing (JERIC)*, 1 (3es), Fall 2001.

[10] Stephen Cooper, Wanda Dann, and Randy Pausch, “Teaching objects-first in introductory computer science”, In *Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE ’03)*. ACM, New York, NY, USA, pp 191-195, 2003.

[11] J. Ryoo, F. Fonseca, D.S. Janzen, “Teaching Object-Oriented Software Engineering Through Problem-Based Learning in the Context of Game Design”, *IEEE 21St Conference on Software Engineering Education and Training*, pp 137-144, Charleston, South Carolina, 2008

[12] Kay, R., “Examining the Effectiveness of Web-Based Learning Tools in Middle and Secondary School Science Classrooms”, *Interdisciplinary Journal of E-Learning and Learning Objects* 7 (1), pp. 359-374, 2011

[13] Tucker, Bill. *The flipped classroom.* Education Next Vol. 12.1: 82-83. 2012

[14] Pat Byrne and Gerry Lyons. “The effect of student attributes on success in programming.” In *Proceedings of the 6th annual conference on Innovation and technology in computer science education (ITiCSE ’01)*. ACM, New York, NY, USA, 49-52. DOI=10.1145/377435.377467 http://doi.acm.org/10.1145/377435.377467, 2001

[15] *tddc32battleship*, last accessed May, 2014 https://code.google.com/p/tddc32battleship/

[16] Teresa Busjahn and Carsten Schulte, “The use of code reading in teaching programming”, In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research (Koli Calling ’13)*. ACM, New York, NY, USA, pp 3-11, 2013

[17] Abrahams I, Millar R., “Does practical work really work? A study of the effectiveness of practical work as a teaching and learning method in school science”, *International Journal of Science Education*;30(14) pp 1945Ð69, 2008

[18] Roberts, Sir G., *SET for success. The supply of people with science, technology, engineering and mathematics skills.* London: HM Treasury, 2002. Last accessed May, 2014 http://webarchive.nationalarchives.gov.uk/+/http:/www.hm-treasury.gov.uk/d/ro

[19] Millar, R, “Rhetoric and reality: What practical work in science education is really for.” In *Practical work in school science: Which way now?*, Editor, J. Wellington, Routledge, London, pp 16-31, 2002

[20] *Javał Platform, Standard Edition 7 API Specification*, Last accessed May, 2014 http://docs.oracle.com/javase/7/docs/api/

[21] McKeithen, K.B., Reitman, J.S., Rueter, H.H.,and Hirtle, S.C. “Knowledge organization and skill differences in computer programmers.” *Cognitive Psychology*, 13, 307-325. 1981

Transforming Computing Education Through Integrated Learning: A 3D Programming Course For Undergraduate Students

Bowu Zhang[1], **and Mira Yun**[2]
[1]Department of Mathematics and Computer Systems, Mercyhurst University, Erie, PA, USA
[2]Department of Computer Science & Networking, Wentworth Institute of Technology, Boston, MA, USA

Abstract— *Computing is increasingly used in a wide variety of fields, i.e., to compose music, to analyze literature and to detect the century-old remains in archeology research. Those interests in computing have led to a growing global trend in the college education where students and faculty are engaged in computer-based study of human-computer interaction, digitization, advanced visualization and other research techniques. This paper explores the integrated learning in computing and other non-computing disciplines by looking at an undergraduate course of 3D programming. Students in this course can experience programming through multiple lenses so that they can have a comprehensive understanding of not only programming concepts, but also the knowledge of applying the computing tools to issues in their own fields. In particular, we demonstrate a number of student projects from this course to show that this integrated learning equips students with valuable tools and knowledge that will build their skill sets, enhance their careers and prepare them for complicated issues facing the world today.*

Keywords: Computing Education, 3D Programming, Integrated Learning

1. Introduction

Computing has played a significant role in a wide variety of fields including mathematics, biology, chemistry, art, music and history [1]–[7]. Nowadays almost every field incorporates substantial work and research involving scientific computation [8], data processing and analysis [9], and software design [10], etc. For example, biological scientists may need computing to analyze the protein structure in a human cell [11], while archeology professors may use computing to estimate the age of historic remains [12]. Today, computing enables and inspires new ways to create and interact in every aspect of our lives, while those non-computing fields inform how we might use new technology to enrich our human experience. Therefore, the computing courses constantly attracts many students who are not CS majors but have passion for learning useful computing tools to support interests in their primary fields. However, traditional computing courses have been referred by many students as abstract, difficult, and boring [13], thus many gave up in the middle or just stayed away from computing courses.

In this paper, we discuss the integrated learning in computing and other disciplines. We keep track of the progresses in an undergraduate computing course to see how we are able to make the learning of computing courses enjoyable by integrating interesting topics from a variety of fields. The integration will allow students to gain knowledge from distinct fields. Integrated learning is the core component of today's college education [14]–[16]. In this constantly changing world, high-end education should prepare students for a life where everything is increasingly interconnected. In other words, students should have a comprehensive understanding of not only their own field, but also a broad knowledge of the changing world. We will look at one undergraduate computing course to display that how the integration complements a variety of other majors, including Anthropology, Biochemistry, Biology, Chemistry, Earth Sciences, Physics, Psychology, and Sociology. The class of "Introduction to 3D programming" is a 3-credit course for non-cs majors at the Mercyhurst University. This course is designed to provide a broad exposure to the field, increasing employability in a variety of settings related to computing. Similar to most of the computing courses, "Introduction to 3D programming" includes a final project where students make contacts with practitioners in their field, complete a research project, and present their work. We will show that courses such as the 3D programming equip students with valuable tools and knowledge that will enhance their careers or prepare them for continued study in graduate and post-baccalaureate professional programs. These give our students the opportunity to cross the technology divisions, so that students will get ready for the transdisciplinary challenges they will face in the world ahead.

The rest of the paper is organized as follows. Section 2 presents the most related work. Sections 3 discusses how the integrated learning is implemented in a 3D programming course for undergraduate programs. A number of course projects are detailed in section 4, and evaluated in terms of integration with non-computing disciplines. Finally, we conclude our work in section 5.

2. Related Work

The computing education has attracted attentions from educational professionals of all stages including the early childhood development, the elementary education, the secondary education and the high education, as computing is increasingly used in every division of our lives. Various methods have been discussed to stimulate student interest in learning computing courses. Based on the course topics, previous research on computing education can be divided into two categories: 1) research on traditional computing courses including programming, software design and data analysis [17]–[19]; and 2) research on integrated computing courses which integrates computing with other disciplines such as art, history and biology [20]–[22]. Work on the first category mainly aim to simplify and visualize the abstract concepts in hard core computing courses. Authors in [23] suggested to add more user-friendly interfaces in programming environment to make users/students comfortable with programming. In [24], authors evaluated the popular programming languages used in education, and they concluded that the object-oriented language shows superiority over other languages in computing education, as students can easily relate the concepts to their lives. A variety of new courses and new education tools have been proposed in the research of the second category. [25] introduced a new major named the art in computing where music, visual art, and theatre are involved in the computing education, and they reported that this new major significantly increased participation in computing, especially attracted students who are not necessarily strong in mathematics. A 3D mapping course in Geology was introduced in [26]. The authors demonstrated a 3D virtual geology field trip and discussed the development of the 3D educational environments. Similar courses have been brought in by [27], [28] where cutting-edge techniques are employed to motivate students learning in computing courses. In this paper, we focus on the integrated learning in computing education. A 3D programming course is analyzed to demonstrate how different disciplines can be integrated into computing. Student learning outcomes support that this integrated learning equips students with valuable tools and knowledge that will build their skill sets, enhance their careers and prepare them for complicated issues facing the world today.

3. Teaching 3D Programming in Alice

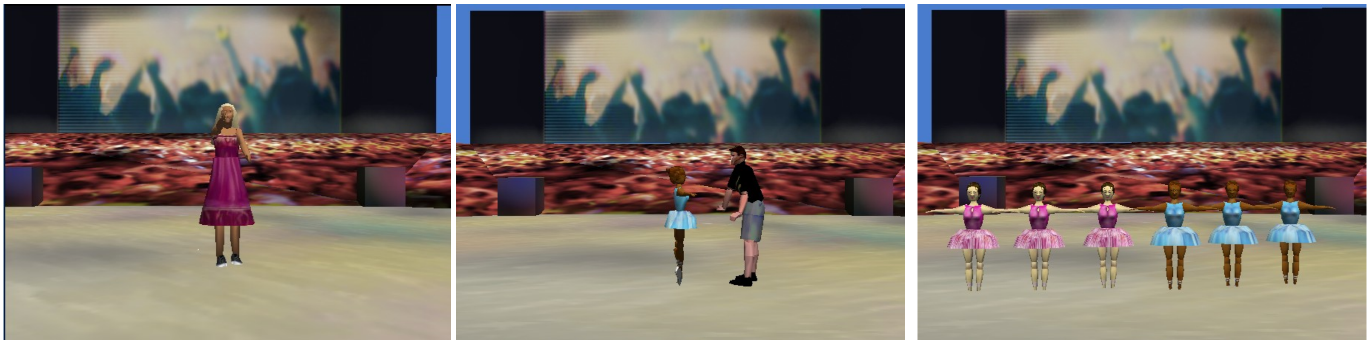
The course of ”Introduction to 3D Programming” is a 3-credit class at the Mercyhurst University. The majority of student population come from mathematics, business intelligence, music and dance. A substantial group of students do not have any prior programming experiences. In the course, we study the 3D programming in Alice. Alice is a open-source programming environment to create animated movies and videos. Different from the traditional programming languages such as c, c++, and java, that emphasize the semantics heavily, Alice adopts a virtual world where users can simply ”drag-and-drop” instructions to create programs without worrying about syntax errors. In Alice, every item is modeled as a 3D object which can be placed into the virtual world, and every object has built-in functions and properties that can be used in programs to let them perform operations that users want. Users can view how the program is executed through the animated movie, resulting in a straightforward understanding of how the programming concepts are applied to objects in the virtual world. Users also can access sounds, images, and texts, enabling them to learn programming from varied perspectives. In addition, users can create their own objects and functions and share with the entire Alice community. In a word, Alice makes the programming easy and interactive, and therefore is widely used in the introductory level programming courses [29], [30].

The programming topics discussed in this course include objects, variables, functions, selection structure, loop structure, and mouse and keyboard events. Typically, students have opportunities to practice every programming concept taught by creating animated movies and simple video games. Exercises about objects include Alice and Rabbit [31]. Exercises about variables include simple mathematic operations. Exercise about selection structure include the grading problem [31]. Exercises about loop structure include the repetitive song of crazy scientist [31]. Exercises about mouse and keyboard events include the helicopter simulation problem [31]. A final project is required at the end of semester that integrates aspects of all programming concepts discussed and demonstrates understanding of the synthesis among them.

Many efforts are made to set up the integrated learning in 3D programming and other seemingly unrelated disciplines. The integration begins with the spirit of object oriented programming, which constitutes the foundation of the Alice software. Since everything is made visualized in the Alice virtual world, it is easy for students to relate the programming to their personal lives. The examples and exercises used in the class are also based on the real world scenarios such as daily talk, gaming, and shopping, which show students the possibility to model every aspect of our life in 3D programming. The final project is designed to frame opportunities for students, after exploring course offerings, to begin a work they may never consider before. Topics ranging from dance choreography to biology DNA pattern recognition allow students to tailor the programming to fit their interests. Compared with that of other programming courses (Java, C++ or Python), student satisfaction with the course of 3D programming is around %10 higher. This course gains the general popularity among non-CS major students.

4. Course Work

The course of ”Introduction to 3D Programming” emphasizes hands-on exercises heavily. Programming exercises under all kinds of scenarios are assigned after learning every new concept. Those exercises help students build the skills and allow students to explore the increasing connections between the programming and the life/field they are familiar with. More importantly, students will need to complete a final project that integrates aspects of all discussed concepts and demonstrates understanding of the synthesis among them. Various topics ranging from dance choreography to biology DNA pattern recognition have been chosen in the past years. An interesting observation is that students tend to tailor the programming to fit their interests in their primary fields. Music majors often prefer to include complicated sounds effects in their projects, while mathematic majors like to



(a) (b) (c)

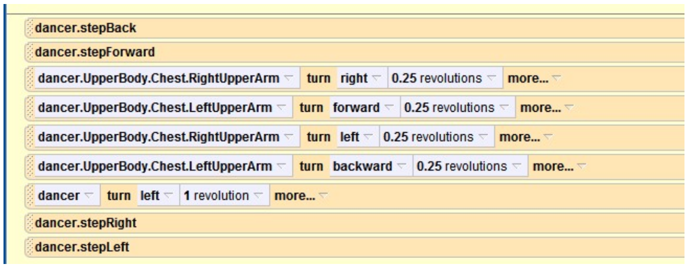
Fig. 1
(A) Individual Dance; (B) Pair Dance; (C) Group Dance.

play with statistics (e.g., random number question/quick math problem). Students are motivated to apply what they learn about programming to topics they deal with most of the time in college. Those projects significantly enhance their understanding in both programming and their own fields. A number of examples projects are discussed as shown in the following sections.

4.1 Dance Choreograph Project

Music majors and dance majors tend to dance choreography projects. Dance movements in the projects are often designed based on the real dance programs and implemented by manipulating the body of human figures in Alice. Also movements need to be adapted and timed to fit the music. In addition, to make the dance ”flow”, transitions are often included. The dance project often involves a number of figures including human and animals dancing on a stage. Control structures such as the selection and the loop statements are often used. The basic program flow is described as follows: There is a group of dancers on the stage. The dancers begin to dance when the music start, and they will stop once the music ends. We classify the dance projects into two categories based on the number of dancers: 1) individual dance where each dancer dances independently; 2) group dance where dancers are implemented as a collection, and perform similar dance movements simultaneously or in turns. Detailed implementations are displayed below.

First, let’s focus on the individual dance program. The program begins with adding a dancer to the virtual world from the local gallery. We may need to adjust the camera so that the dancer is right in front of the camera. Then in the object tree, select the dancer and create an object method named ”dancing”. Next, add instructions to the dancing method to let the dancer move. We may pose the legs, arms and other parts of body to create different movements. For example, we can let the arm swing by dragging and dropping the statement ”dancer-upper-body-left arm.turn right 1/2 revolution”, or we can let the dancer spin by the statement ”dancer.turn right 1 revolution”. Between movements, we let the camera zoom in and zoom out to make a special visual effect using the dummy objects: We drop invisible dummy objects at the position where we want the camera to be; Then let the camera move by specifying the target as the created the dummy object. Once we’ve made a step or two that we are pleased with, we can make our dance method repeat itself by using a loop. Suppose we have a dance method that works nicely for a step or two as in figure 2



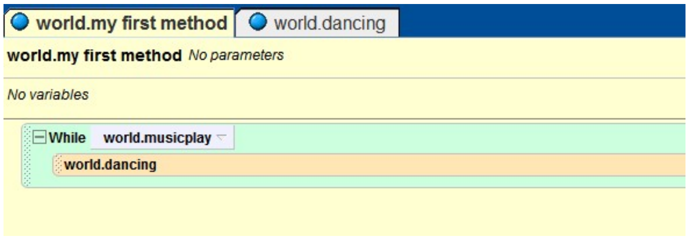
```
dancer.stepBack
dancer.stepForward
dancer.UpperBody.Chest.RightUpperArm  turn right  0.25 revolutions  more...
dancer.UpperBody.Chest.LeftUpperArm  turn forward  0.25 revolutions  more...
dancer.UpperBody.Chest.RightUpperArm  turn left  0.25 revolutions  more...
dancer.UpperBody.Chest.LeftUpperArm  turn backward  0.25 revolutions  more...
dancer  turn left  1 revolution  more...
dancer.stepRight
dancer.stepLeft
```

Fig. 2
Individual Dancing Method

To make it look like a real dance, we need to use some randomness. The world has random functions that can produce random numbers within a fixed range. We can let the dancer move a random distance forward or backward or let him/her spin a random number of revolutions. Test the method by dragging it to the world.my first method. Go back to the dancing method and let the dancer repeat the routine movements by using the loop. Here we can specify a fixed number of iterations or have the dance method keep repeating over and over while the music plays. One way to make the dance repeat indefinitely is to call the dancing method within a loop in the my first method. All we need do is to add two lines in the world. my first method 3

To insert music into the program, we can use the object.play() method to import existing songs or you can use recorder to make your own dance music!

Group dancing requires synchronized movements. For example, dancers need to turn right simultaneously, or as some

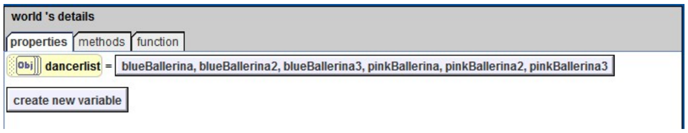


```
world.my first method   No parameters
No variables
⊟While   world.musicplay
    world.dancing
```

Fig. 3

REPETITION IN WORLD.MY FIRST METHOD

dancers move a leg forward, the others should move the other leg back, and so on. An easy way to implement the group dancing is to put all dancers in a special data structure called list. Lists are data structures used to group and to organize objects. Since list items can be processed one-by-one or all at once (The control structures for all in order and for all together correspond to these), by using lists, we can save on writing code if we want to process similar actions on a group of objects. What we need is to write a single dancing method and apply it to each list item in turn rather than to write a method for each item or having a separate call for each item. We can access a particular object by its index in the list. To create a list of dancers, we need to add a number dancers to the virtual world (Fig. 4).



```
Obj dancerlist = blueBallerina, blueBallerina2, blueBallerina3, pinkBallerina, pinkBallerina2, pinkBallerina3
```

Fig. 4

A LIST OF DANCERS

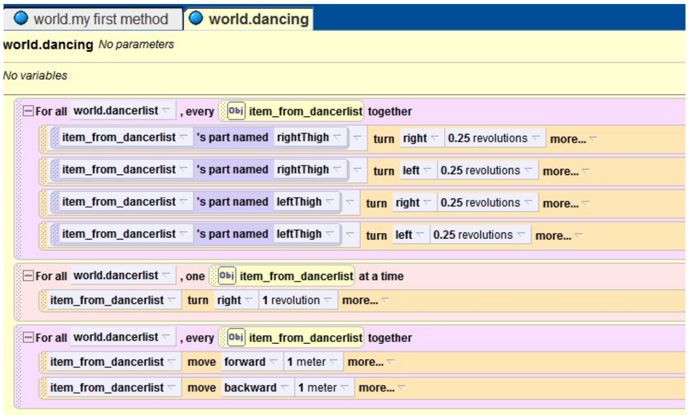
Similar to the individual dancing method, in the group dancing method, we just need to create a few steps. We can do with as few as Fig. 5 and 6(a)

To apply the above the dancing method to the list, we need to either call ”for every item in the list all together” or ”for every item in the list all in order” as shown in the figure 6(a). Check the dancing method to make sure there are no errors. Then we can call the dancing method in the my first method repeatedly to fit with the music.

Special stage light and camera movement can be adopted to create magic stage effects. We may also insert images or pictures of friends/families to make the program more fun.

4.2 Soccer Player Project

Mathematic majors and sport team players often dive into sports related projects. In these projects, Alice is used to model a common scene on the field, such as a ice hockey game or base ball game. A typical example comes from the soccer ball game where the user controls a soccer player to shoot the goal. Projects that fall in this category are very



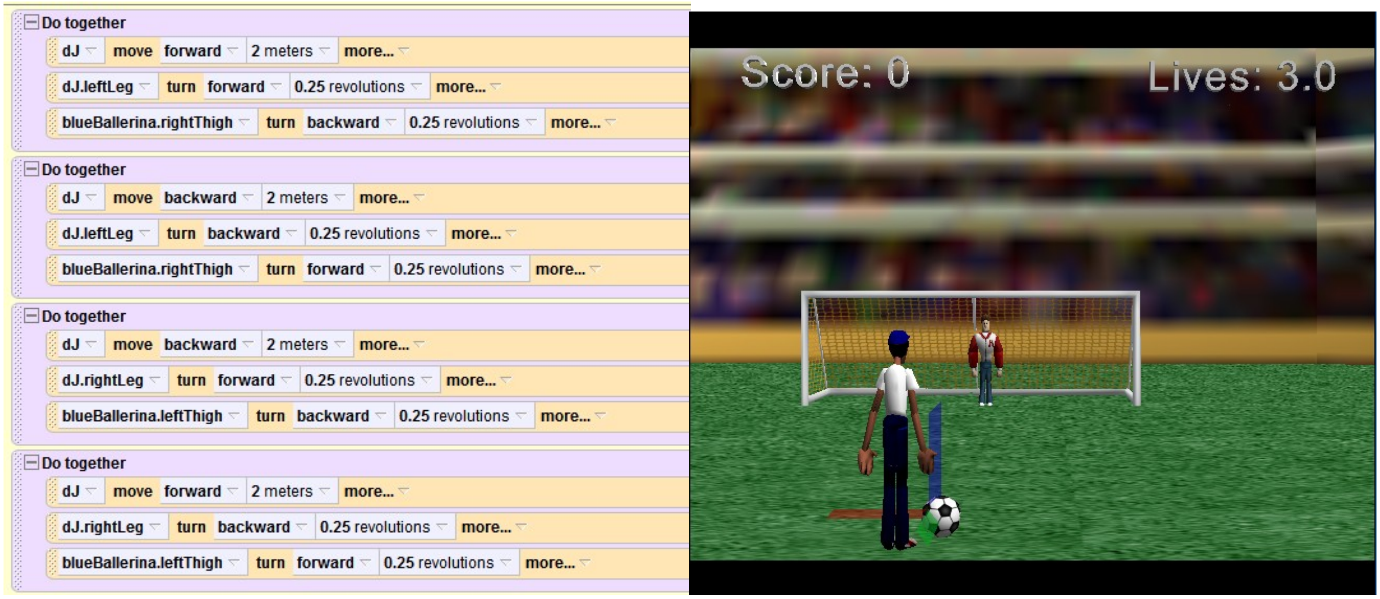
```
world.dancing   No parameters
No variables
⊟For all world.dancerlist , every Obj item_from_dancerlist together
    item_from_dancerlist 's part named rightThigh   turn right 0.25 revolutions more...
    item_from_dancerlist 's part named rightThigh   turn left 0.25 revolutions more...
    item_from_dancerlist 's part named leftThigh   turn right 0.25 revolutions more...
    item_from_dancerlist 's part named leftThigh   turn left 0.25 revolutions more...
⊟For all world.dancerlist , one Obj item_from_dancerlist at a time
    item_from_dancerlist turn right 1 revolution more...
⊟For all world.dancerlist , every Obj item_from_dancerlist together
    item_from_dancerlist move forward 1 meter more...
    item_from_dancerlist move backward 1 meter more...
```

Fig. 5

DANCING METHOD OF TWO DANCERS

interactive users can enjoy playing the game by the keyboard or the mouse while a score object is used to keep track of the progress he or she has made. The basic program flow is described as follows: There is(are) a (or a number of) soccer player(s) and a goal keeper on the soccer field. The soccer player is controlled by a set of particular keys or the mouse to shoot the goal. The user can determine the direction the ball kicked into and the speed that the ball will travel at, while the direction and the distance the goal keeper jumps to defend his goal is controlled by the computer. Player's statistics are often considered in the game, i.e, the chance a soccer player uses his left foot to shoot the goal or the average distance that a ball can travel after kicked by a particular player, which leads to variations to the game. Random functions are often employed to reflect the sport statistics, such as the direction the player may kick the ball or the height the goal keeper to jump. Here we use the random number generator in Alice to implement the randomness in this game. In order to let player kick the ball to the user specified direction, we first need to place a number of invisible circles (Assume there are N circles) which serve as targets within the gate. Then we assign each circle a unique number ranging from $[1, N]$ as its ID so that every circle represents a direction where the player can shoot the gate. What the program does is to generate a random integer on a scale of 1 to N, and let the soccer player shoot the circle whose ID is equal to the random number generated. Or the user can click on one of circles to trigger the mouse click event which make the ball fly to that circle. The random number is also used on the goal keeper to prevent the player from scoring. Normally, the goal keeper can jump in 3 (or 5) directions: up, left and right (upper left, upper right), each of which is represented by an integer from $[1, 3]$. Then the program every time generates a random integer within $[1, 3]$ to indicate the direction the goal keeper to move, and another random number to indicate the height to jump.

Detailed implementations are displayed below. First, add a player and a goal keeper to the virtual world. Adjust their



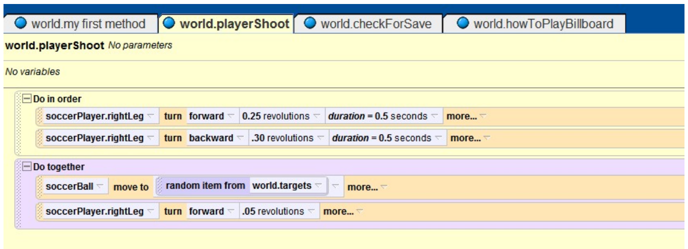
```
Do together
    dJ  move  forward  2 meters  more...
    dJ.leftLeg  turn  forward  0.25 revolutions  more...
    blueBallerina.rightThigh  turn  backward  0.25 revolutions  more...
Do together
    dJ  move  backward  2 meters  more...
    dJ.leftLeg  turn  backward  0.25 revolutions  more...
    blueBallerina.rightThigh  turn  forward  0.25 revolutions  more...
Do together
    dJ  move  backward  2 meters  more...
    dJ.rightLeg  turn  forward  0.25 revolutions  more...
    blueBallerina.leftThigh  turn  backward  0.25 revolutions  more...
Do together
    dJ  move  forward  2 meters  more...
    dJ.rightLeg  turn  backward  0.25 revolutions  more...
    blueBallerina.leftThigh  turn  forward  0.25 revolutions  more...
```

(a) (b)

Fig. 6

(A) DANCING METHOD OF A LIST OF DANCERS; (B) SOCCER PLAYER.

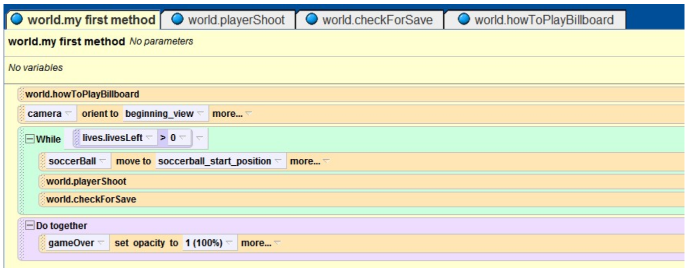
initial positions to put them right in front of the camera. Then drop a dummy object to label the starting position of the ball so that the ball can be placed back for the next game every time it is kicked off. Create a world level method named ”playershoot” to let the user shoot the goal. In the new method, we may let the user use the mouse or the keyboard to choose a circle to kick the ball (invisible circle). Manipulate the body of the player to perform the action of shooting. Let the ball move to the goal(circle) while let the keeper jump simultaneously to stop the ball. The method may look like Fig. 7



```
world.playerShoot  No parameters

No variables

Do in order
    soccerPlayer.rightLeg  turn  forward  0.25 revolutions  duration = 0.5 seconds  more...
    soccerPlayer.rightLeg  turn  backward  .30 revolutions  duration = 0.5 seconds  more...
Do together
    soccerBall  move to  random item from  world.targets  more...
    soccerPlayer.rightLeg  turn  forward  .05 revolutions  more...
```

Fig. 7

THE PLAYERSHOOT METHOD

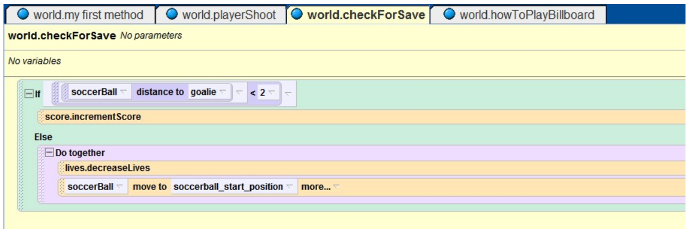


```
world.my first method  No parameters

No variables

world.howToPlayBillboard
camera  orient to  beginning_view  more...
While  lives.livesLeft  > 0
    soccerBall  move to  soccerball_start_position  more...
    world.playerShoot
    world.checkForSave
Do together
    gameOver  set opacity to  1 (100%)  more...
```

Fig. 8

THE WORLD. MY FIRST METHOD OF SOCCER PLAYER PROGRAM

We want the user to play this game a number of times. Therefore, we add statements in the world.my first method to call the ”playershoot” method repetitively (Fig. 8). On the other hand, we create an object to keep track of scores the user has got and display the value in the virtual world. Thus another world level method is created to detect if the goal keeper catches the ball. The value of the score object will be incremented by one if the ball is not catched. An example method is shown in the figure below:



```
world.checkForSave  No parameters

No variables

If  soccerBall  distance to  goalie  < 2
    score.incrementScore
Else
    Do together
        lives.decreaseLives
        soccerBall  move to  soccerball_start_position  more...
```

Fig. 9

KEEPING SCORE OF THE SOCCER PLAYER

This project can be easily modified to a multiple-player game by assigning different players different keyboard events. More statistical patterns can be applied to the multiple-player game since the there might be many variations to the collaborations among players. Professional soccer games such FIFA World Game are based on the same set of design principles. Note that the more players there are in the field, the more random the result will be. This is consistent with the probability theory that the number of possible cases increases as the input size increases. On the other hand, multiple story threads can be implemented in this project by adding the selection control structures. However, this needs careful designs of logic to differentiate the conditions for each possible result.

5. Conclusion

In this paper, we explore the integrated learning in computing courses in undergraduate programs. Integrative courses encourage students to explore important topics through multiple lenses so that they understand how to approach complicated issues facing the world today. In particular, we looked at one programming course for non-cs majors. Course projects were analyzed to show that integrated learning enable students to create their own work in a thoughtful way so they can explore their interests, build skill sets, engage their intellects and learn from a variety of fields.

References

[1] D. Blank, J. S. Kay, J. B. Marshall, K. O'Hara, and M. Russo, "Calico: a multi-programming-language, multi-context framework designed for computer science education," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. ACM, 2012, pp. 63–68.

[2] V. Ananthanarayanan and W. Thies, "Biocoder: A programming language for standardizing and automating biology protocols," *Journal of biological engineering*, vol. 4, no. 1, pp. 1–13, 2010.

[3] R. Libeskind-Hadas and E. Bush, "A first course in computing with applications to biology," *Briefings in bioinformatics*, vol. 14, no. 5, pp. 610–617, 2013.

[4] J. C. Rubinstein, "Perspectives on an education in computational biology and medicine," *The Yale journal of biology and medicine*, vol. 85, no. 3, p. 331, 2012.

[5] J. Bresson, C. Agon, and G. Assayag, "Openmusic: visual programming environment for music composition, analysis and research," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 743–746.

[6] W. Scacchi, "The future of research in computer games and virtual worlds," *Institute for Software Research, University of California, Irvine, Irvine*, 2012.

[7] R. Wille, S. Offermann, and R. Drechsler, "Syrec: A programming language for synthesis of reversible circuits," in *System Specification and Design Languages*. Springer, 2012, pp. 207–222.

[8] P. Wallisch, M. E. Lusignan, M. D. Benayoun, T. I. Baker, A. S. Dickey, and N. G. Hatsopoulos, *MATLAB for neuroscientists: an introduction to scientific computing in MATLAB*. Academic Press, 2013.

[9] F. Erickson, "Qualitative research methods for science education," in *Second international handbook of science education*. Springer, 2012, pp. 1451–1469.

[10] Y. Cai, D. Iannuzzi, and S. Wong, "Leveraging design structure matrices in software design education," in *Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on*. IEEE, 2011, pp. 179–188.

[11] E. H. Kellogg, A. Leaver-Fay, and D. Baker, "Role of conformational sampling in computing mutation-induced changes in protein structure and stability," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 3, pp. 830–838, 2011.

[12] Y. Kondo, S. Kadowaki, H. Kato, M. Naganuma, A. Ono, K. Sano, and Y. Nishiaki, "Network computing for archaeology: a case study from the replacement of neanderthals by modern humans database project," *Revive the Past*, p. 217, 2011.

[13] R. Pau, W. Hall, and M. Grace, "Its boring: female students experience of studying ict and computing," *School Science Review*, vol. 92, no. 341, pp. 89–94, 2011.

[14] L. D. Fink, *Creating significant learning experiences: An integrated approach to designing college courses*. John Wiley & Sons, 2013.

[15] M. A. Honey, M. Hilton, *et al.*, *Learning science through computer games and simulations*. National Academies Press, 2011.

[16] J. Duffy, L. Barrington, C. West, M. Heredia, and C. Barry, "Service-learning integrated throughout a college of engineering (slice)." *Advances in Engineering Education*, vol. 2, no. 4, 2011.

[17] C. A. Shaffer, T. L. Naps, and E. Fouh, "Truly interactive textbooks for computer science education," in *Proceedings of the 6th Program Visualization Workshop*, 2011, pp. 97–103.

[18] J. B. Fenwick Jr, B. L. Kurtz, and J. Hollingsworth, "Teaching mobile computing and developing software to support computer science education," in *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011, pp. 589–594.

[19] M. Sahami, M. Guzdial, A. McGettrick, and S. Roach, "Setting the stage for computing curricula 2013: computer science–report from the acm/ieee-cs joint task force," in *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011, pp. 161–162.

[20] A. Schäfer, J. Holz, T. Leonhardt, U. Schroeder, P. Brauner, and M. Ziefle, "From boring to scoring–a collaborative serious game for learning and practicing mathematical logic for computer science education," *Computer Science Education*, vol. 23, no. 2, pp. 87–111, 2013.

[21] L. Ma, J. Ferguson, M. Roper, and M. Wood, "Investigating and improving the models of programming concepts held by novice programmers," *Computer Science Education*, vol. 21, no. 1, pp. 57–80, 2011.

[22] J. B. Labov, A. H. Reid, and K. R. Yamamoto, "Integrated biology and undergraduate science education: a new biology education for the twenty-first century?" *CBE-Life Sciences Education*, vol. 9, no. 1, pp. 10–16, 2010.

[23] O. Shaer and E. Hornecker, "Tangible user interfaces: past, present, and future directions," *Foundations and Trends in Human-Computer Interaction*, vol. 3, no. 1–2, pp. 1–137, 2010.

[24] M. S. Horn, E. T. Solovey, R. J. Crouser, and R. J. Jacob, "Comparing the use of tangible and graphical programming languages for informal science education," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 975–984.

[25] B. Manaris, R. McCauley, M. Mazzone, and W. Bares, "Computing in the arts: a model curriculum," in *Proceedings of the 45th ACM technical symposium on Computer science education*. ACM, 2014, pp. 451–456.

[26] S. J. Whitmeyer, "Community mapping in geology education and research: How digital field methods empower student creation of accurate geologic maps," *Geological Society of America Special Papers*, vol. 486, pp. 171–174, 2012.

[27] S. Bogaerts, K. Burke, and E. Stahlberg, "Integrating parallel and distributed computing into undergraduate courses at all levels," in *First NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar-11), Anchorage, AK*, 2011.

[28] Y. Khmelevsky and V. Voytenko, "Cloud computing infrastructure prototype for university education and research," in *Proceedings of the 15th Western Canadian Conference on Computing Education*. ACM, 2010, p. 8.

[29] W. P. Dann, S. Cooper, and R. Pausch, *Learning to Program with Alice (w/CD ROM)*. Prentice Hall Press, 2011.

[30] T. Daly, "Minimizing to maximize: an initial attempt at teaching introductory programming using alice," *Journal of Computing Sciences in Colleges*, vol. 26, no. 5, pp. 23–30, 2011.

[31] J. Adams, *Alice in Action: Computing Through Animation*. Cengage Learning, 2006.

Offering Service Learning Projects in a New Environment

Donald R. Schwartz
School of Computer Science and Mathematics
Marist College
Poughkeepsie, NY 12601

Abstract - *The focus of service learning is to provide opportunities for our students to gain real, practical experience while completing "real world" projects, which often become extensions of the classroom itself. One of the major goals of service learning is to enhance student learning while at the same time meeting a need within the local community "beyond the gates" of our College. This paper briefly describes the challenges of starting up Service Learning projects at a new College and how I dealt with those challenges. The remainder of the paper describes several of the Service Learning projects my students completed during my first semester at the new school.*

Keywords: Team Projects and Case Studies, Collaborative Learning, Service Learning, Projects and Software Engineering

1 Introduction

After many years of very successfully incorporating Service Learning projects into my courses, I found myself in a new situation. My former school's administration decided to eliminate the Department of Computer Science (which I chaired), so I suddenly found myself looking for a new job. I knew that my preference was to find school that embraced the Liberal Arts and had a strong Computer Science program. Happily, I was successful in securing a position at such a school!

I found myself "starting over" in many areas, with new colleagues, new students, a completely new environment, new courses, new schedules – basically, new "everything". My initial inkling was to wait a while before attempting to tackle Service Learning projects in the midst of all of this "newness", but I decided to "go for it" and see what happened. As it turns out, what happened was very good!

2 Getting (Re)Started

In this new environment, I no longer had my well-established list of contacts throughout the community, my stockpile of recently-proposed projects that hadn't yet been assigned (along with eager, built-in clients), my reputation of a successful history of past projects, or my standard request-for-proposals email messages. So, I wrote up a short email message (since it was the Friday before the first week of classes and everyone was being inundated with the usual beginning-of-the-school-year email barrage) in which I introduced myself as the new guy in Computer Science, briefly described the types of projects I was seeking, and asked for help in spreading the word:

> Subject: Any "Computer Projects" on your wish list?
>
> Hi, all.
>
> Please forgive the "mass email" nature of this message, but I figured it was the "safest" way to reach interested parties.
>
> I am teaching Software Development II this semester and am looking for good, interesting semester projects for the students to work on. We are looking for "real world" projects that will benefit on- and/or off-campus groups. For off-campus groups, our preferences are non-profits and K-12 education.
>
> I'm not really looking for tiny projects -- something more substantial would be better. For example, in the past my students have written software for the Mississippi Department of Health, Mississippi Children's Home Services, several local public and private schools (learning "games" for pre-K students, math quiz "games" for 1st - 6th grade). If you're not sure whether your project is the "right" size, just contact me and we can discuss it.
>
> I usually send out this information well in advance of the semester, but since I'm "that new guy in Computer Science", this is a last-minute request.
>
> My plan is for the students in my classes to choose a set of projects for each class to work on next Thursday and Friday (Sept. 5th and 6th), so it would be great if we could discuss your project ideas by Wednesday, the 4th, at the latest.
>
> Sooooo, do you have any computer application needs for your area? If you do, please reply and let me know.
>
> Thanks!
>
> --Don

I wasn't sure what to expect in terms of responses, but happily was overwhelmed by the number of replies that came in – several dozen in the first day! I met with many people who had wonderful ideas and tried to help them create brief descriptions of their projects so that I could present them to my classes.

My arrival in my new department coincided with the second year of its newly-revised curriculum. Students begin their studies with CMPT120 – Introduction to Programming, which exposes them to introductory programming skills using HTML, CSS and JavaScript for program development. The second required course is CMPT220 – Software Development I, in which students continue to master software development skills while using Java and JavaScript to learn about basic data structures. CMPT221 – Software Development II is the third required course our students take. In this course, students are exposed to client-side programming, server-side programming and data integration over the web. Our development environment is PHP and MySQL.

Software Development II seemed to be a nice “fit” for larger projects in which students worked in dynamic groups throughout the Software Engineering Life Cycle. My normal approach is to have students work in teams to create the deliverables for each Life Cycle phase (requirements, specifications, preliminary design, detailed design, implementation and testing) for each project. At each milestone, the students change projects and change teams. This has the advantage of exposing students to different projects at different stages with different teammates, which is likely to be what most students will experience out in the “real world” after graduation. The process I use for managing these class projects has been described in [1][2][3][4][5].

The proposed projects were presented as possible semester-long projects. The students in my Software Development II courses were also required to propose semester projects for the class to consider. After all of the ideas were presented, the students themselves selected the projects we would tackle. The classes conducted client interviews for each project, then teams took each project though each milestone. At the end of the semester, the clients attended presentations for each project. The remainder of this paper describes several of the Service Learning projects that my classes completed during the Fall 2013 semester.

3 The Projects

3.1 Department of Corrections

Our client requested the development of a website to track data about the Transition Program within the Dutchess County Sheriff’s Office Department of Corrections Division. [6] The Transition Program is an invitation-only program for inmates who are approaching their release dates and have maintained a good record while serving their time. Inmates who participate in the program have their own housing unit, recreation yard and counseling system. Participants are offered special outreach programs related to domestic violence, parenting, substance abuse, anger management, creative writing and film therapy. Those who do not have a high school diploma (or equivalent) are required to work toward their GED while in the Transition Program. All of these efforts are aimed at reducing recidivism and easing the transition back into the wider society. Contact does not end after an inmate is released; the Transition Program continues to work with the participants, helping them find assistance they might need, contacting them monthly to check up on them, and helping to ensure they are meeting any

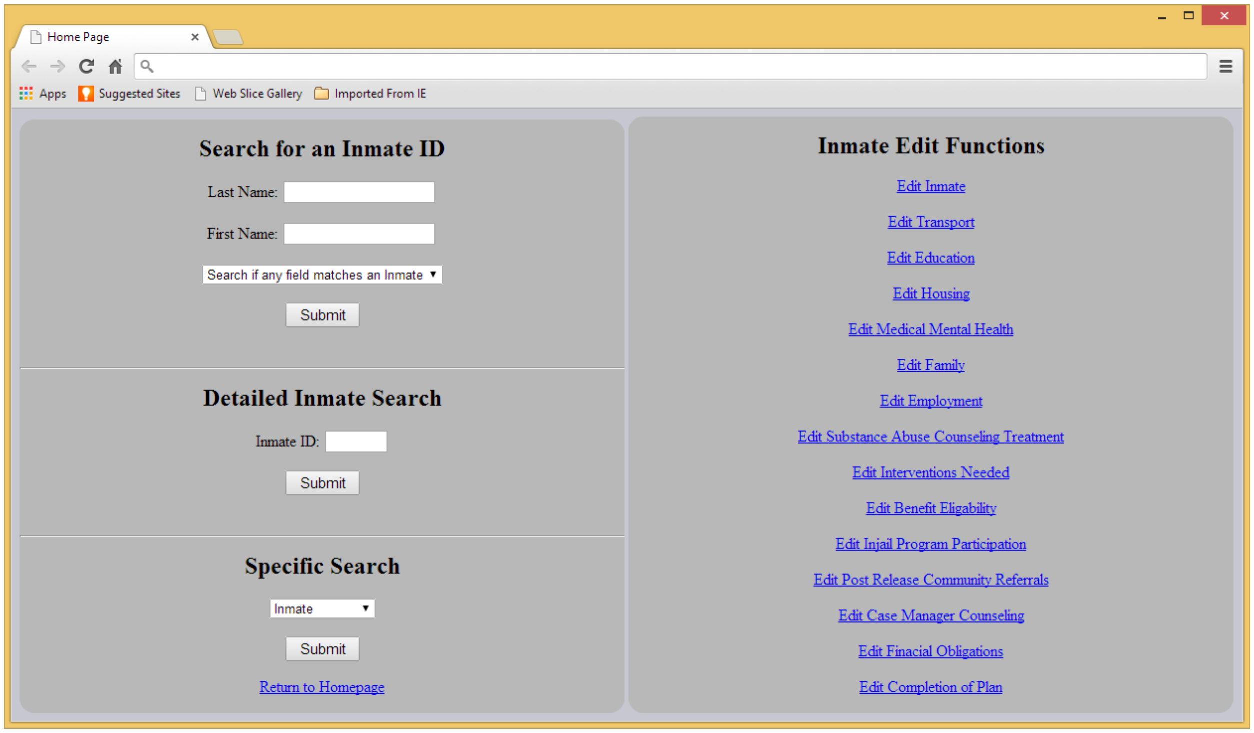


Figure 1 - Department of Corrections Search Screen

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

required post-release requirements that might have been assigned by the Courts.

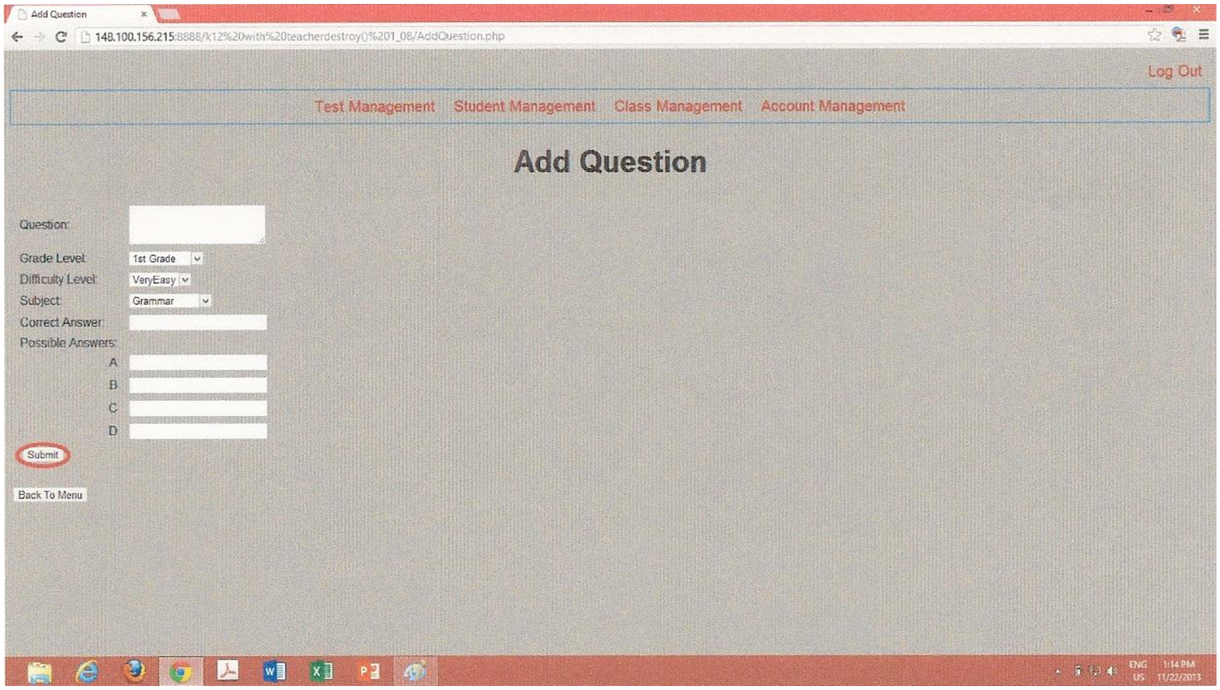
The Department of Corrections has its own data system for its entire inmate population. Of course, this system is not set up for the specialized needs of the Transition Program (whether the participants will be eligible for public assistance, Medicaid, SSI or the like; will have a place to live, transportation, and employment; have special medical or mental health needs; need post-release community referrals, and so on), so most the additional data tracking and recording for the Program was being done on paper – not an ideal situation, to say the least. We received copies of all of the paper forms that were being used and developed a website that could be used by the Program officers to track the data, run queries and generate reports.

3.2 Internship Tracker

Our client for this project was one of our Computing Technology faculty members who is in charge of managing all of the internship opportunities for our Computer Science, Information Science and Information Technology majors. (Our internship program has been tremendously successful, and tracking all of this data has become quite an undertaking.) Features requested for the proposed website include managing each of our internship and career fairs, tracking all of the companies/organizations that offer internships (along with descriptions of each internship, their requirements and their pay ranges), the students who have completed internships (and the feedback they were willing to provide about their experiences), and a straight-forward means to allow current students to research the available internships for which they meet the requirements. The ability to generate reports and statistics for members of the administration were also included.

3.3 Practice Tests for the New York State 3rd Grade Reading Examination

This client, a school system located in the suburbs of New York City, wanted a system that would expose its Kindergarten, first- and second-grade students to the online environment they would experience when taking the New York State Department of Education English Language Arts 3rd Grade Examination. [7] Their current practice is to give students sample multiple-choice tests on paper, but they wanted the students to be able to take these practice tests online to more realistically represent the actual testing environment. We were tasked with the development of a system that would allow teachers to create sample questions and compile them into grade-appropriate tests (with the ability to have the system shuffle the order of the multiple-choice options presented for each question); grade those tests and report the results by student, class, and grade level; and offer feedback about the students' progress, including questions or areas where students may have struggled.



Add Question

Question:

Grade Level: 1st Grade

Difficulty Level: VeryEasy

Subject: Grammar

Correct Answer:

Possible Answers:

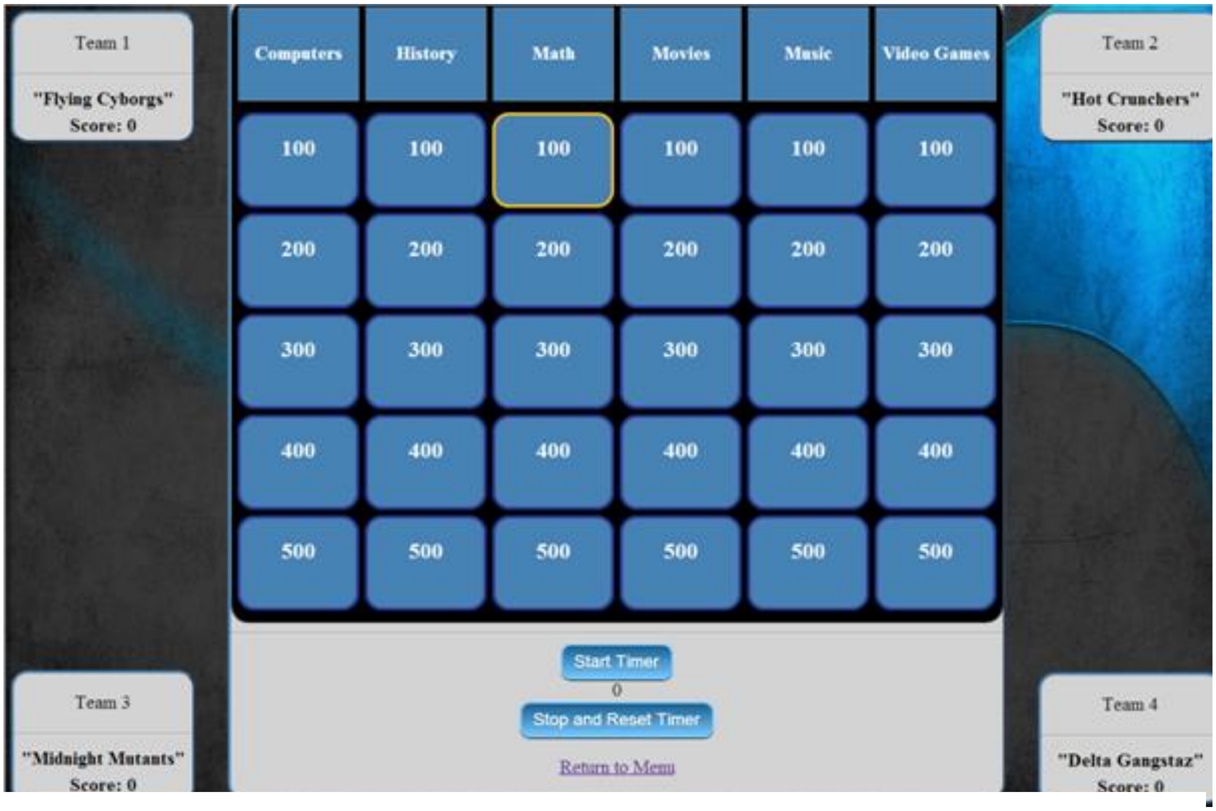
A

B

C

D

Figure 2 - Practice Test Generator - Add Question Screen



Team 1
"Flying Cyborgs"
Score: 0

Team 2
"Hot Crunchers"
Score: 0

Computers	History	Math	Movies	Music	Video Games
100	100	100	100	100	100
200	200	200	200	200	200
300	300	300	300	300	300
400	400	400	400	400	400
500	500	500	500	500	500

Team 3
"Midnight Mutants"
Score: 0

Team 4
"Delta Gangstaz"
Score: 0

Figure 3 - Jeopardy Main Screen

3.4 Jeopardy

A colleague who teaches business classes submitted a proposal for an in-class review/quiz system based on the *Jeopardy!* game show. This client developed a version of this system in PowerPoint, but was unhappy with its static nature. The proposed website would allow the professor to create new topic categories, enter multiple questions and answers for each difficulty level within each category, and create student teams as contestants for the game. To start a new game, the professor selects categories, after which the system generates a new game board with randomly-selected question/answer pairs, along with a randomly-selected "double jeopardy" indicator.

3.5 Fang Protective Services

Fang Protective Services is a constituent organization of the U.S. State Department's Overseas Security Council of Diplomatic Security. Their mission is to enhance "the safety and security of faith-based humanitarian and medical mission teams as they care for the most vulnerable members of humanity".[8] They offer training and seminars to various non-governmental organizations (NGOs), help with the development of security policies and crisis management plans, and conduct risk assessment activities.

Their project proposal included the development of a website that would provide instruction-based materials and security-based briefings for their clients, as well as training and testing features for security personnel. The instructional materials include documents, reports and videos. The training and testing materials include written and video-based quizzes and simulations. The video quizzes are designed to take users through "real world" scenarios, occasionally stopping the demos to quiz the user about what to do next. The action continues (or not) based on the user's responses.

3.6 Center for the Prevention of Child Abuse

The Center for the Prevention of Child Abuse is a local non-profit agency whose mission is "to reduce the incidence of child abuse and neglect by providing education, prevention, and intervention services throughout Dutchess County". [9] They provide various educational parenting services, including personal safety programs, community education and mandated reporters training, parent empowerment programs, special needs parenting programs and teen parenting programs. They are a private company that receives some funding from various federal and state agencies, but they also depend heavily on donations from various private and community sources. As such, they are often asked to provide detailed reports about the levels of success they achieve with the various programs they offer. Their current practice is to collect program-specific data using paper forms and Excel spreadsheets. Their proposal requested a website that would provide a much easier platform for data collection and reporting. They provided us with a

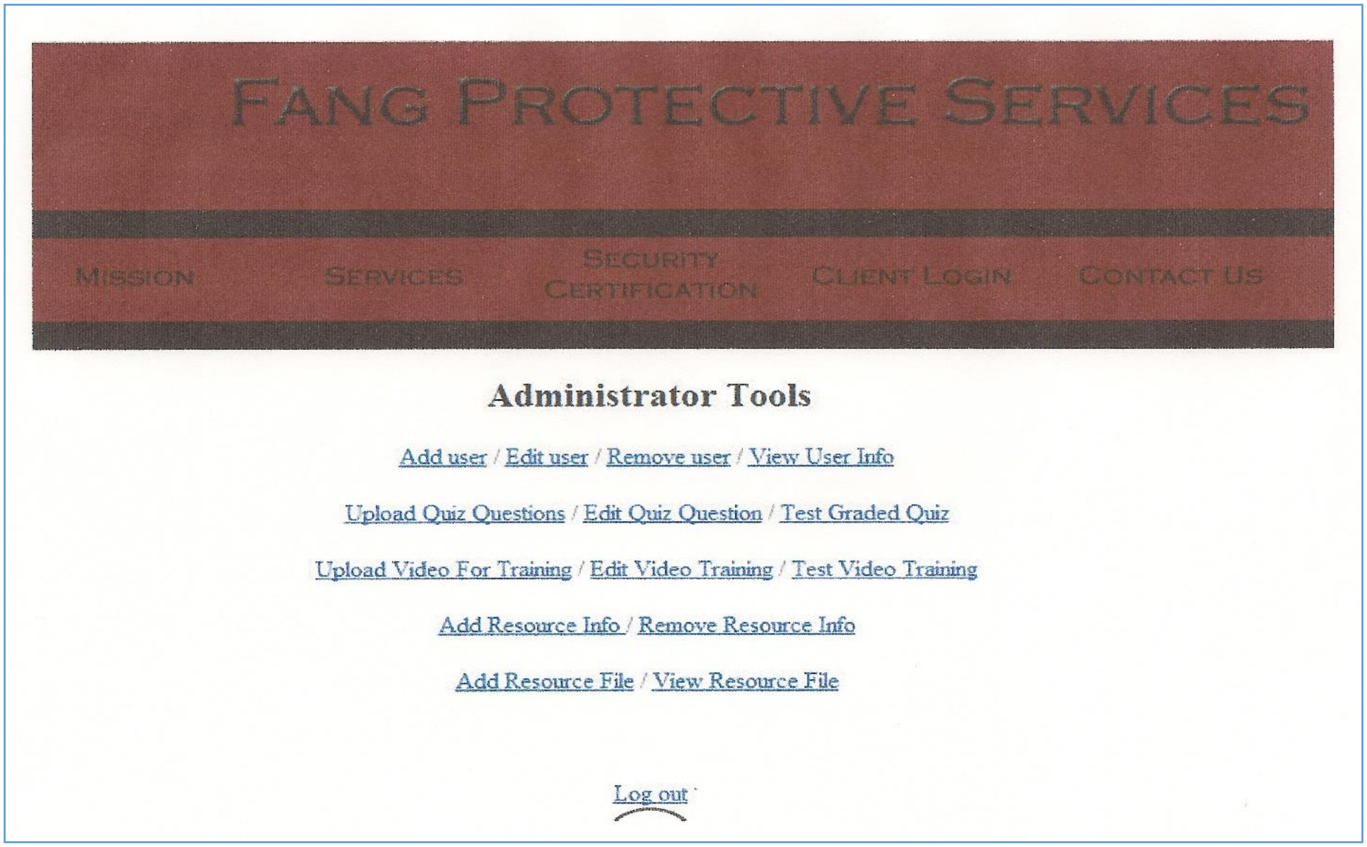

FANG PROTECTIVE SERVICES
MISSION
SERVICES
SECURITY CERTIFICATION
CLIENT LOGIN
CONTACT US
Administrator Tools
Add user / Edit user / Remove user / View User Info
Upload Quiz Questions / Edit Quiz Question / Test Graded Quiz
Upload Video For Training / Edit Video Training / Test Video Training
Add Resource Info / Remove Resource Info
Add Resource File / View Resource File
Log out

Figure 4 - Fang Protective Services Administrator Tools

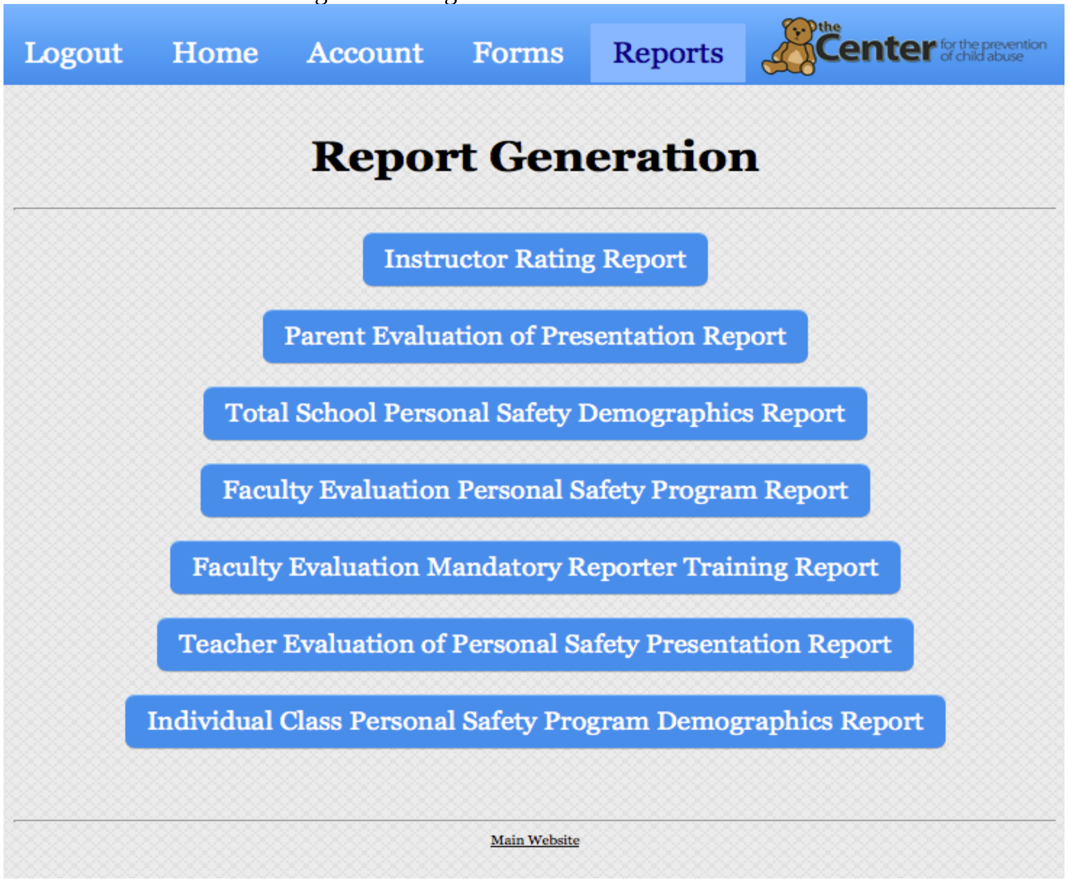

Logout
Home
Account
Forms
Reports
the Center for the prevention of child abuse
Report Generation
Instructor Rating Report
Parent Evaluation of Presentation Report
Total School Personal Safety Demographics Report
Faculty Evaluation Personal Safety Program Report
Faculty Evaluation Mandatory Reporter Training Report
Teacher Evaluation of Personal Safety Presentation Report
Individual Class Personal Safety Program Demographics Report
Main Website

Figure 5 - Center for the Prevention of Child Abuse Report Generation Page

collection of forms and spreadsheet examples, which we then converted into web pages and connected to a centralized database.

3.7 Clinical Rotation

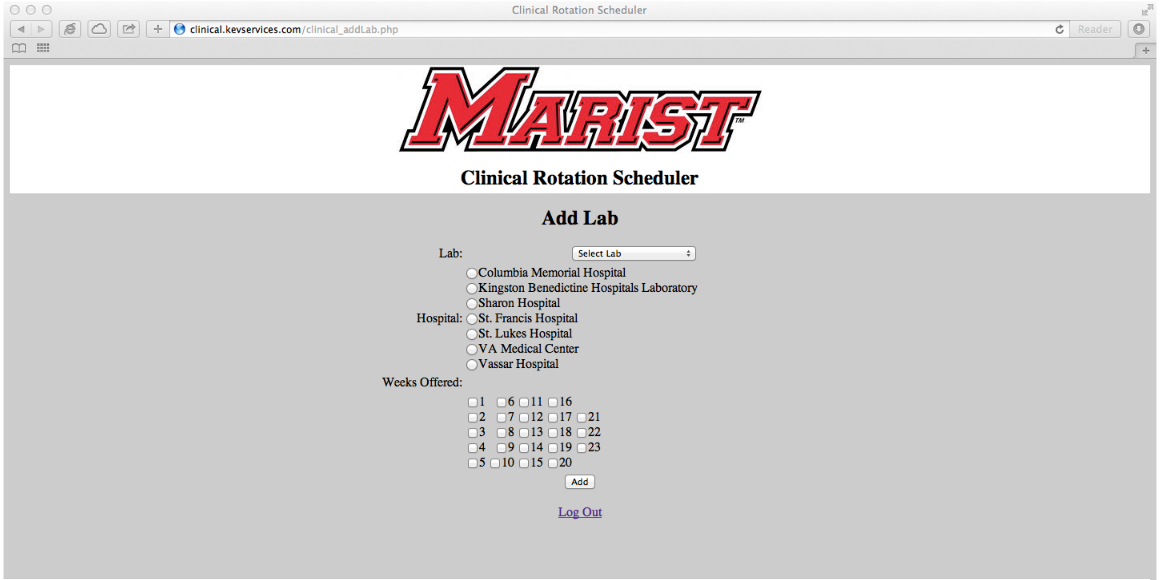
Our college's nationally accredited Medical Technology Program is a cooperative effort that includes eight local hospitals and clinical laboratories in the surrounding area. The program requires that students complete a 23-week clinical rotation as part of the requirements to earn a B.S. degree with a major in Medical Technology. The number of students participating in each rotation period is capped at 24. The rotation includes the following lab requirements: five weeks in Hematology, five weeks in a Blood Bank, five weeks in clinical chemistry, six weeks in clinical microbiology, one week in urinalysis, and one week in outpatient phlebotomy followed by three weeks of inpatient phlebotomy. Only the inpatient phlebotomy lab may overlap with another rotation at the same location. Only one student is allowed in each lab at each location, with one exception: one of the hospital locations allows for two students to concurrently complete one, or sometimes two, of the labs. Most, but not all, of the locations offer every lab requirement. Most, but not all, labs at each location are available during the entire 23-week period of the rotation.

The director of the program had been using a paper-based system (involving multi-color pens and flags) to schedule the 24 students in the various labs, so she was thrilled to be able to propose a project aimed at automating this system. The system would allow directors to indicate which hospitals and clinics were participating in the program, specify the labs being offered at each location, enter the weeks the labs were available, and input information about each student in the rotation. The system would then create schedules for each student and lab location and provide those schedules to each student and hospital/clinic. The directors would also receive master schedules with full details about everything.

The clients for this project were probably the most thrilled of all clients with the finished product. However, in the interest of full disclosure, I must report that the students (remember, this is a sophomore-level class) were unable to successfully tackle the scheduling algorithm, so I had to step in and develop a solution. My initial thought was to simply write an algorithm that recursively generated all possible schedules until it found one that worked. However, that approach did not work since each particular student's labs could start and finish on any week of the rotation. The other complicating factor was the issue with the in-patient phlebotomy lab, since it could overlap with any other lab (that the student had not yet completed) at the same location as that student's out-patient phlebotomy lab. This required that the algorithm include a look-ahead feature to determine whether there were any future labs (that the student had not yet completed) available at that location. Coming up with that code in a short period of time was indeed quite challenging (and may likely become the focus of a publication in the near future!).

4 Conclusions

Service Learning projects offer a wide variety of benefits. Students get experience interacting with "real world" clients while developing "real world" projects that will actually benefit many constituents. Our clients benefit by receiving no-cost, top-quality software tailored especially to their needs. Outreach into the wider community can help to strengthen ties between the academic institution and its local neighbors. Reaching out to these neighbors, especially if you are the "new kid in town" can seem daunting, but it does not have to be. Take a chance – I'm very happy that I did!


Clinical Rotation Scheduler
clinical.kevservices.com/clinical_addLab.php
Reader
MARIST

Clinical Rotation Scheduler

Add Lab

Lab: Select Lab

Hospital:
○Columbia Memorial Hospital
○Kingston Benedictine Hospitals Laboratory
○Sharon Hospital
○St. Francis Hospital
○St. Lukes Hospital
○VA Medical Center
○Vassar Hospital

Weeks Offered:

□1 □6 □11 □16
□2 □7 □12 □17 □21
□3 □8 □13 □18 □22
□4 □9 □14 □19 □23
□5 □10 □15 □20

Add

Log Out

Figure 6 - Clinical Rotation Add Lab Page

5 References

[1] Schwartz, Donald R., “Beyond Programming and Software Development: Additional Teaching/Learning Goals throughout the Curriculum”, Proceedings of the 2011 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-11), Las Vegas, July 2011.

[2] Schwartz, Donald R., “Incorporating Service Learning into Capstone/Software Engineering Group Projects”, Proceedings of the 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-10), Las Vegas, July 2010.

[3] Schwartz, Donald R., “The Benefits (and Drawbacks!) of “Real World” Software Engineering Semester Projects: A Case Study”, Proceedings of the 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS-09), Las Vegas, July 2009.

[4] Schwartz, Donald R., “Service Learning and Software Engineering: Managing ‘Real World’ Group Projects”, Proceedings of the 2007 International Conference on Software Engineering and Data Engineering (SEDE-07), Las Vegas, July 2007.

[5] Schwartz, Donald R., “Managing and Assessing Software Engineering Group Projects”, Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition, June 2005.

[6] http://www.co.dutchess.ny.us/CountyGov/Departments/Jail/JLIndex.htm

[7] http://www.nysedregents.org/ComprehensiveEnglish/

[8] http://www.fangprotectiveservices.org/fps.html

[9] http://www.preventchildabusedutchess.org/

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Using a Web-Based Testing Tool Repository in Programming Course: An Empirical Study

Anurag Goswami
North Dakota State University
Computer Science Department
Fargo, USA
anurag.goswami@ndsu.edu

Gursimran S. Walia
North Dakota State University
Computer Science Department
Fargo, USA
Gursimran.walia@ndsu.edu

Sameer Abufardeh
North Dakota State University
Computer Science Department
Fargo, USA
Sameer.abufardeh@ndsu.edu

ABSTRACT

This paper highlights an important issue of the knowledge and skill deficiency of software testing among undergraduate students in software engineering discipline. The paper provides an approach for integrating software testing into computer programming course in a non-obtrusive manner. The paper describes the use of the *Web Based Repository of Software Testing Tools* (WReSTT) that can assist the instructors in integrating the testing component into their software engineering course and also provides the students with all the necessary resources (tutorials, quizzes, videos etc) for them to gain general testing knowledge, be able to apply the testing techniques, and become proficient in the usage of testing tools. This paper presents the design of the WReSTT, and then presents an empirical study that was conducted in an introductory computer programming course at North Dakota State University. The results from the study showed that the WReSTT can be used to significantly impact the testing knowledge gained by the students and that the increased use of the WReSTT resulted in a better grade for the students on their programming assignments.

Categories and Subject Descriptors

D.2.5 [**Testing and Debugging**]: *Testing tools (e.g., data generators, coverage testing).*

General Terms

Measurement, Experimentation, Languages.

Keywords

WReSTT, Software Testing, Unit Testing, Code Coverage, Empirical Study.

1. INTRODUCTION

Software testing continues to be a concept that plagues software engineering students when they enter the software industry because after their undergraduate degrees. On that note, previous researchers have found evidence that the graduating software engineering students and the newly hired software engineering lack basic testing knowledge, testing ability, and the usage of software testing tools [1-4, 8]. This has been identified both by the industrial researchers and the academics as an important knowledge and skill deficiency among the software engineering students. More specifically, the researchers have shown through studies conducted with senior-level undergraduate students that the students do not have the knowledge of different testing techniques, lacked the ability to use testing tool (with code coverage in particular) and created ineffective test cases [1, 2, 3, 4, 8]. While the previous research does not provide or highlight a specific type of testing the student's lack, it is believed that the testing as a whole is an important knowledge deficiency in graduating students and is a focus of attention of this research paper.

While there have been approaches to integrate testing into computer programming courses at various institutions, they do not rigorously enforce the testing in these course especially during the introductory computer programming courses. Furthermore, when testing is used (e.g., upper level courses), there is not enough support for the students to expose them to the testing tools and assist them during the usage of testing tools on their programming assignments. While a lot of testing content is being taught in the upper level courses or the graduate level courses at our institution, there is a lack of focus of integrating testing early in the curriculum. This paper describes the use of the *Web Based Repository of Software Testing Tools* (WReSTT) that can assist the instructors in integrating the testing component into their software engineering course. The WReSTT also provides the students with all the necessary resources (tutorials, quizzes, videos etc) for them to gain general testing knowledge, be able to apply the testing techniques, and become proficient in the usage of testing tools.

The WReSTT also incorporates a collaborative and social networking environment where the students enrolled in a class can communicate with each other, start and contribute to the discussion of software testing related topics, gain and compete for the virtual points (e.g., the number of times each student visited the testing tutorial) for testing related work, and provide greater student involvement, cooperation, and team work through the allocation of the virtual points. The students using the WReSTT can browse different tutorials depending upon the programming language employed in their course.

This paper presents the design of the WReSTT, its main features, and the integration of WReSTT in software engineering courses without affecting the logistics of the course. The paper presents an empirical study that was conducted in an introductory computer programming course at North Dakota State University. The goal of the study was to evaluate the impact of WReSTT on the undergraduate students acquisition of the knowledge of testing objectives, testing techniques, their usage of testing tools, and their proficiency on using the testing tools (in particular the Unit testing and the Code Coverage). The results from the study showed that the WReSTT can be used to significantly impact the testing knowledge gained by the students and that the students are highly likely to no use any other resource or online learning resource if they are not exposed to the WReSTT in their courses. The results also showed that the increased use of the WReSTT

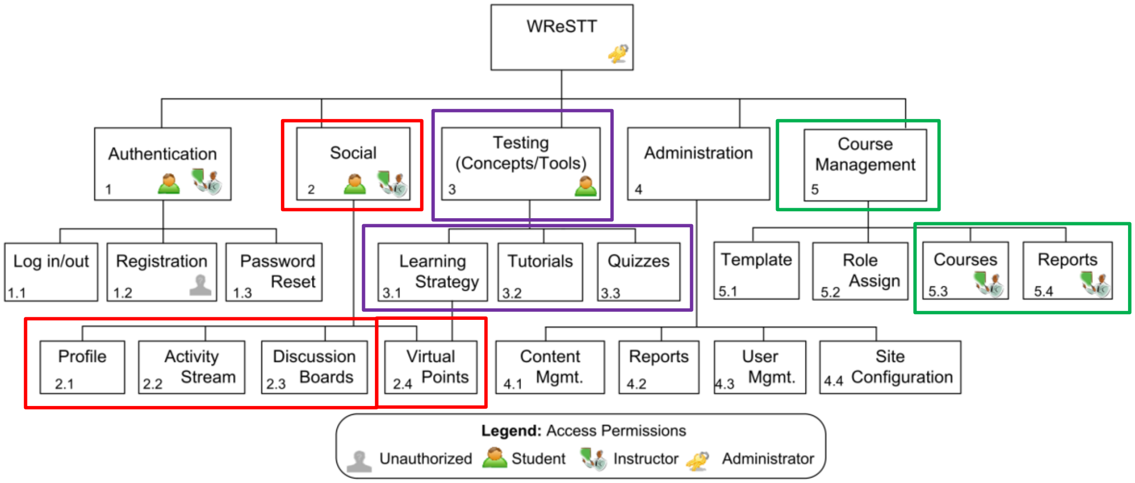


Figure 1. Design of WReSTT [5, 9]

resulted in a better grade for the students on their programming assignments.

The rest of the paper is organized as follows. Section 2 describes the main features of the WReSTT, along with the instructor and the student view of the WReSTT. Section 3 describes the experiment design, the participating students, and the data collected during the experiment run. Section 4 presents the analysis of the data organized around the study goals. The discussion of results is provided in Section 5 followed by the concluding remarks.

2. WEB-BASED REPOSITORY OF SOFTWARE TESTING TOOLS (WReSTT)

This section provides an introduction of the WReSTT and its main features to expose undergraduate students to the testing methods and tools. The following subsection discusses how the WReSTT can be used in programming courses by the instructors (or TA's) and by the students.

2.1 Introduction to WReSTT: Main Features

The WReSTT is designed to be able to provide online learning resources in software testing. The main components of WReSTT are shown in Figure 1 [5, 9] and discussed as: 1) WReSTT provides the students with *reading tutorials* on a variety of software testing concepts and methodologies. Students browse through the testing tutorials, and evaluate their understanding of those concepts through quizzes. 2) WReSTT also provides students with access to the *video tutorials* on different testing tools classified by *category* (Coverage, Metrics, Plug-ins, Test execution, Web), by *language* (C++, Java, VB .Net), and by *test level* (System/UI, Unit). 3) WReSTT provides a *collaborative* and *social networking* learning environment for the students. WReSTT allows students to upload their user profile, post comments on the discussion board, and monitor the activity of other students enrolled in the same class. A unique feature of WReSTT incorporated assignment of "*virtual points*" to the students (e.g., for completing a tutorial and its quiz, posting a testing related comment related to the forum etc) in order to increase student involvement at the individual level as well as at the class level (by allowing students to monitor the virtual point leaders in the same class).

2.2 Using WReSTT: Instructor View

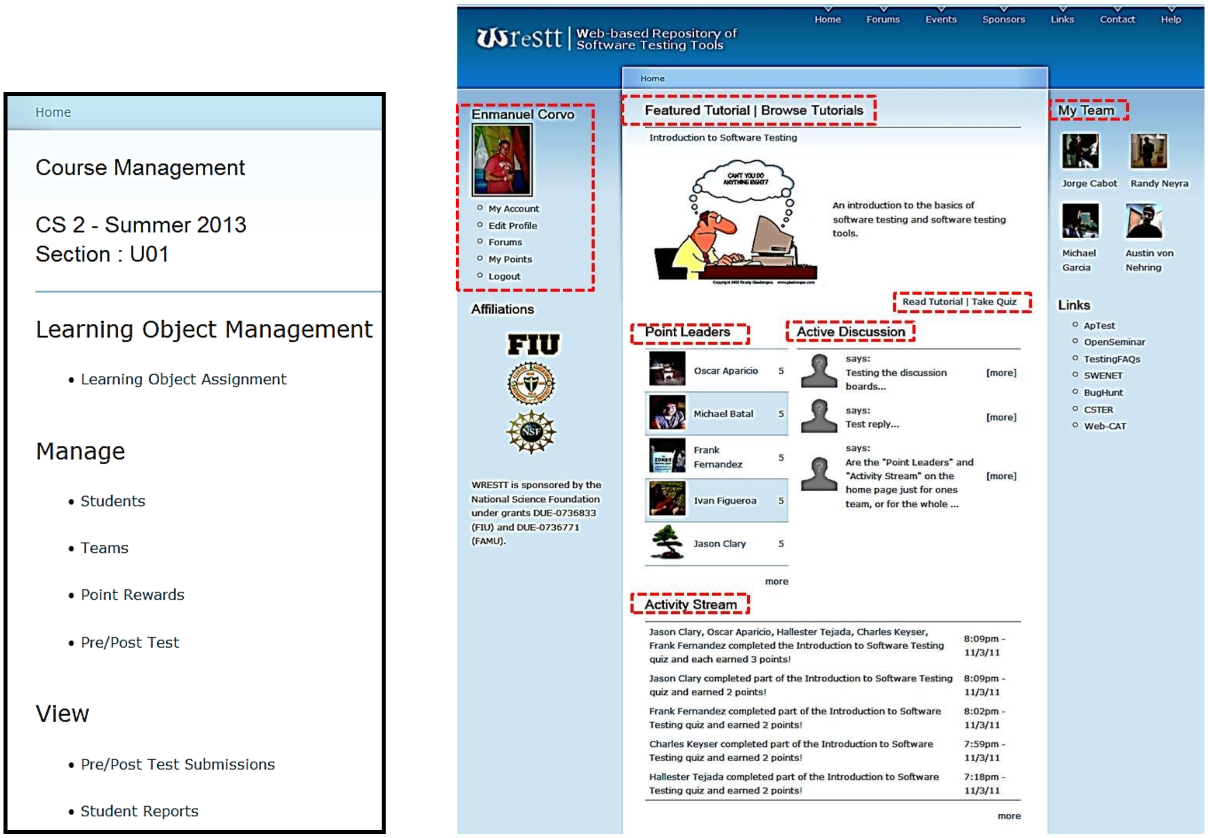
Instructors can use the WReSTT in their courses (that have some programming component) by *creating a course* (e.g., uploading class roster and the assignment of unique ID and password to each student) and through the *course management* (e.g., monitoring the activity streams, generating student reports, and allocation of virtual points for student activities). Figure 2 shows a selection of screen capture (to keep anonymity) that depicts the instructor's interface during the course *creation* and the course *management*.

Course creation: Instructors can use WReSTT at their campus by requesting the access from the administrators (authors) of the WReSTT (link), who will provide the login credentials to the instructor and will create the template for their course title. Next, instructors can login to access their course and use the template instructions to: 1) upload the class roster; 2) create unique login credential for the students; 3) assign students to virtual teams; 4) describe the rubric for the allocation of virtual points for different student activities; 5) enable/disable pre and posttest. There is Help menu available on the WReSTT to assist the course creation.

Course management: After the creation of the course, the instructor can manage the course by: 1) monitoring each student's activity on the WReSTT (e.g., number of times each student visited the tutorial, the time it took them to complete a particular quiz); 2) monitor the virtual points gained by each student in the class; 3) generate and print student reports using different statistics related to the class (e.g., number of students completing a particular quiz, virtual point leaders). Figure 2 shows the instructors' view during the course creation and management.

2.3 Using WReSTT: Student View

As mentioned earlier, students need to authenticate themselves to access the WReSTT for their course. The WReSTT is designed to mirror the social networking tool features to enhance student involvement. Figure 2 (right side) shows the students view after they login to WReSTT.



WRESTT is sponsored by the National Science Foundation under grants DUE-0736833 (FIU) and DUE-0736771 (FAMU).

An introduction to the basics of software testing and software testing tools.

says:
Testing the discussion boards... [more]

says:
Test reply... [more]

says:
Are the "Point Leaders" and "Activity Stream" on the home page just for ones team, or for the whole ... [more]

Jason Clary, Oscar Aparicio, Hallester Tejada, Charles Keyser, Frank Fernandez completed the Introduction to Software Testing quiz and each earned 3 points! 8:09pm - 11/3/11

Jason Clary completed part of the Introduction to Software Testing quiz and earned 2 points! 8:09pm - 11/3/11

Frank Fernandez completed part of the Introduction to Software Testing quiz and earned 2 points! 8:02pm - 11/3/11

Charles Keyser completed part of the Introduction to Software Testing quiz and earned 2 points! 7:59pm - 11/3/11

Hallester Tejada completed part of the Introduction to Software Testing quiz and earned 2 points! 7:18pm - 11/3/11

Fig. 2 Instructor view (left side) and the student view (right side)

Some of the main section of students' interface is highlighted in Figure 2. That is, each student can *create profile*, browse the *testing tutorials*, take *quizzes*, watch videos on the *testing tools*, interact with other students in the class via *testing based discussions*, and monitor the *active discussions*, *activity stream* and the *virtual point leaders*. In addition, depending upon the way an instructor wants to run their course, additional virtual bonus may be assigned for the team-based activities (e.g., all members have to complete a quiz to get team virtual points) to foster team collaboration using WReSTT.

3. STUDY DESIGN

The study was designed to investigate the effect of the WReSTT on the undergraduate students' acquisition of general testing concepts, knowledge of testing techniques, and proficiency of testing tool usage in an introductory computer programming course at North Dakota State University. The study utilized a pretest/posttest design in which the participating subjects were pretested on their testing knowledge prior to the introduction of the WReSTT. Next, the students worked individually to complete programming assignments where they were asked to apply the *JUnit* (a unit testing framework for the Java) and the *EclEmma* (code coverage tool) testing tools using the information contained in the WReSTT. After the completion of the assignments, the subjects were tested again (using the same set of questions in the pretest) to evaluate their difference in the knowledge of the testing concepts and testing techniques and testing tools. At the conclusion of the study, the subjects filled a survey to reflect their experience with the use of WReSTT in the current course and their desire to use WReSTT in future programming classes. More details on the study design are presented in the following paragraphs.

Study Goals: This study has two main goals. The first goal is to analyze the impact the WReSTT had on the student's learning of software testing concepts and tools, and is stated formally as:

Analyze student's pretest and posttest scores *for the purpose of* evaluation *with respect to* the impact of WReSTT on the increase in the knowledge acquisition of testing concepts, techniques, and tools *in the context of* undergraduate students enrolled in programming course at a large public university.

The second goal is to analyze the overall satisfaction with the WReSTT features and its usability in a programming course:

Analyze student's post-study questionnaire response *for the purpose of* evaluating *with respect to* the ease of use and integration of WReSTT in programming course.

Participating Subjects: The participating subjects were 21 undergraduate computer science students enrolled in the Computer Science II'2013 course at North Dakota State University. 18 out of 21 students chose to participate in the study.

Artifacts: The students were asked to submit a written report of the *Unit* and the *Code Coverage* testing tools and techniques used, the test cases, the test execution results during the testing of their code, and their understanding of the results achieved. The artifacts were evaluated by the instructor of the course (not involved during the design of the empirical study).

Study Procedure: The study steps are as follows:

a) *Step 1 – Pretest:* A pretest was administered at the beginning of the study. The test instrument is shown in Appendix A. The goal of the pretest was to measure the student's knowledge of testing concepts and tools prior to using the WReSTT. The questions on the pretest measured the baseline knowledge of the testing concepts, knowledge of testing techniques and testing tools, and the proficiency in testing tools usage.

b) *Step 2 – Training on WReSTT*: Next, the subjects were trained on how to access the WReSTT for browsing tutorials, taking quizzes, posting and viewing discussion threads. Also, the students were taught how to use the WReSTT for watching testing tutorials on different testing techniques and testing tools.

c) *Step 3 – Programming Assignment Description*: Using the information in the WReSTT, the students worked individually on their programming assignments and test their code at the Unit level (using JUnit tool to test classes), and to evaluate the test coverage achieved on their programs (using EclEmma tool). The students used the tutorials on the JUnit and the EclEmma tools to learn how to apply these tools. The students then documented the report of the test execution results along with the source code which was evaluated by the course instructor (not a part of the research team).

d) *Step 5 – Posttest*: At the end of the course assignments, the students were re-tested on their knowledge of testing techniques and tools using the same instruments used during the Step 1 (pretest). The goal of the posttest was to measure the impact of WReSTT on the increase in testing knowledge.

e) *Step 6 – Post-Study Survey:* Finally, the subjects were asked to fill a survey to evaluate the usability and the usefulness of the WReSTT in introductory programming course. The survey questions (shown in Appendix B) included questions related to the student's overall reaction to the WReSTT, questions related to the usefulness and adequacy of the testing tutorials and tools in WReSTT, and open ended questions to gather feedback on improving the usage of WReSTT in future courses .

Data Collection and Evaluation Criterion: The data includes the student's responses on the pretest and the posttest survey. The responses (on pretest and posttest) were evaluated by assigning a value representative of the adequacy of their response. More details are provided in subsection 5.1.

The student's responded to the survey questions using the 5-point likert scale: 1 = Strongly Disagree, 2 = Disagree, 3 = Neither agree nor Disagree, 4 = Agree, 5 = Strongly Agree. We treated these scales as interval scales (rather than the ordinal scales), following the standard practice in the social sciences [ ]. The survey responses for each question were averaged across all the students and evaluated for statistical significance (i.e., whether the average is significantly greater than the middle point of the scale). The subject's responses to open ended questions were collected to help researchers better understand the results. More details are provided in subsection 4.2

We also collected the grade received by students on the testing assignments to correlate their perceived usefulness of WReSTT on their grades.

4. RESULTS AND ANALYSIS

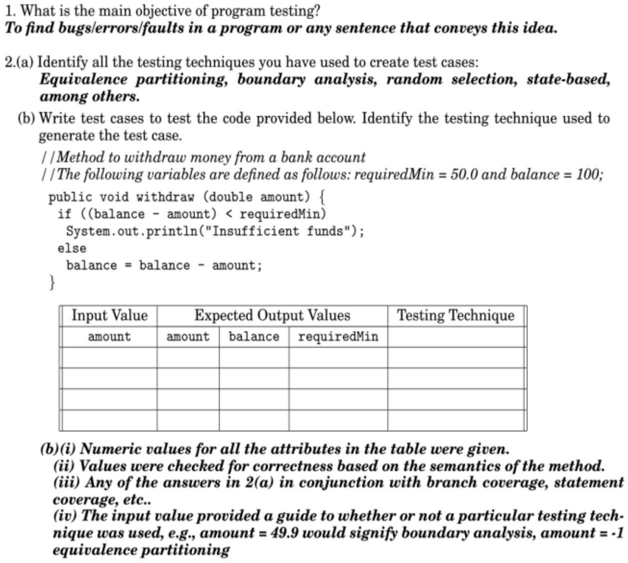
The results are organized around study goals presented in Section 3. Section 4.1 compares the results regarding the students testing knowledge after and before using the WReSTT. Section 4.2 evaluates the students' feedback on the use of WReSTT in programming assignments. In addition, this section presents the correlation between the usefulness of WReSTT against the performance on the assignments.

4.1 Pretest vs. Posttest Results

To evaluate the impact of the WReSTT on the students' knowledge acquisition of the testing concepts and their proficiency of the testing tools and techniques, a comparison of the pre-test and post-test results was performed. The students were assigned a score representative of their responses to the questions on the pretest and posttest. The rubric for the expected answers for each question is shown in Appendix A that was used to calculate the total score for each student. Next, the average score of all the eighteen students on the pretest vs. the average score on the posttest were compared. The result showed that the subjects, performed better on the posttest (an average score of 6.43) as compared to their performance on the pretest (an average score of 13.85). A one-sample ANOVA test showed that the increase in the overall testing knowledge (measured by the increase in the average score of subjects during the posttest when compared to their. Pretest score) after being exposed to the WReSTT is statistically significant ($p < 0.05$).

While this result in interesting, we wanted to evaluate the usefulness of WReSTT seperately on the students' increase in the a) general software testing knowledge, b) testing tool usage, and c) the proficiency level of the testing tools usage.

General testing knowledge: The student responses to the questions 1, 2a, and 2b were evaluated during the pretest and during the posttest. As mentioned earlier, the students were assigned scores for each question based on the correctness of their response (see Figure 3 for the questions and the rubric). Then, the scores on these questions were added to indicate the student's general knowledge of software testing and the testing technique.



1. What is the main objective of program testing?
To find bugs/errors/faults in a program or any sentence that conveys this idea.

2.(a) Identify all the testing techniques you have used to create test cases:
Equivalence partitioning, boundary analysis, random selection, state-based, among others.

(b) Write test cases to test the code provided below. Identify the testing technique used to generate the test case.

//Method to withdraw money from a bank account
//The following variables are defined as follows: requiredMin = 50.0 and balance = 100;

```
public void withdraw (double amount) {
  if ((balance - amount) < requiredMin)
   System.out.println("Insufficient funds");
  else
   balance = balance - amount;
}
```

Input Value	Expected Output Values			Testing Technique
amount	amount	balance	requiredMin	

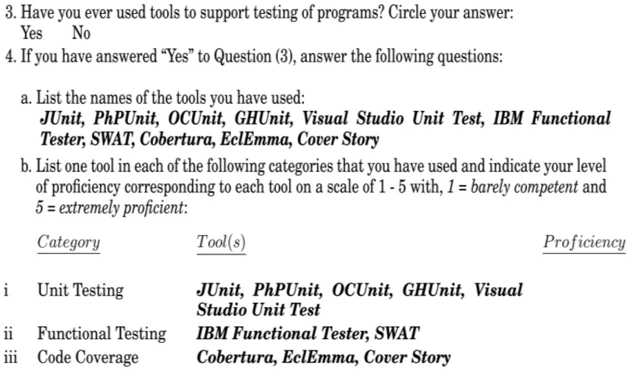
(b)(i) Numeric values for all the attributes in the table were given.
(ii) Values were checked for correctness based on the semantics of the method.
(iii) Any of the answers in 2(a) in conjunction with branch coverage, statement coverage, etc..
(iv) The input value provided a guide to whether or not a particular testing technique was used, e.g., amount = 49.9 would signify boundary analysis, amount = -1 equivalence partitioning

Fig. 3 General testing knowledge question and expected answers

The average scores of the "*general testing knowledge*" of students increased from an average of 0.45 during the pretest to an average score of 2.12 during the posttest. A one-sample ANOVA test demonstrated a significant increase in the general testing knowledge of students (p<0.05) and a significant increase in the number of students that were able to apply at least one particular testing technique (based on the question 2 (b) responses) during the posttest. Therefore, the WReSTT was able to improve the students understanding of general testing concepts and their ability to apply a particular technique to test the code.

Testing tool usage: The questions 3 and 4 (as shown in Figure 4) were used to evaluate the students testing tool knowledge and its usage. The scores for questions 3 and 4 were added to indicate their usage of testing tools. The results showed that, on average, the subjects knowledge and the usage of testing tools was significantly higher during the posttest (an average of 0.58) when compared to their pretest score (an average of 3.2) at p<0.01 level. Furthermore, the students during the posttest were able to identify a significantly larger number of tools for each category (i.e., Unit testing, Functional testing, Code Coverage) when compared to their responses during the pretest.

The students also rated their proficiency levels in using the testing tools (part of question 4(b)). This was done to understand whether the perception of their proficiency for specific tools also increased (in addition to their actual ability). The students rated their proficiency for Unit testing, functional testing, and code coverage on a scale of 1-5 with 1-barely competent and 5-extremely competent. The results showed that, on average, the subjects felt more proficient in the testing tool usage (when averaged across all the three categories for all the students) during the posttest as compared to their proficiency of tools during the pretest. A Paired-sample Wilcoxon Signed-Rank test to compare the average of the ratings at pretest vs. posttest found the increase in the perceived proficiency to be statistically significant (p = 0.012).



3. Have you ever used tools to support testing of programs? Circle your answer:
Yes No

4. If you have answered "Yes" to Question (3), answer the following questions:

a. List the names of the tools you have used:
JUnit, PhPUnit, OCUnit, GHUnit, Visual Studio Unit Test, IBM Functional Tester, SWAT, Cobertura, EclEmma, Cover Story

b. List one tool in each of the following categories that you have used and indicate your level of proficiency corresponding to each tool on a scale of 1 - 5 with, *1 = barely competent* and *5 = extremely proficient*:

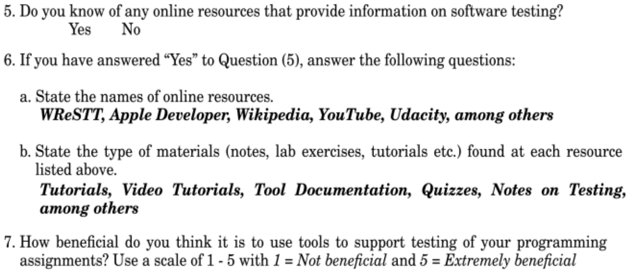
	Category	*Tool(s)*	*Proficiency*
i	Unit Testing	***JUnit, PhPUnit, OCUnit, GHUnit, Visual Studio Unit Test***	
ii	Functional Testing	***IBM Functional Tester, SWAT***	
iii	Code Coverage	***Cobertura, EclEmma, Cover Story***	

Fig. 4 Testing tool usage questions

Next, questions 5, 6, and 7 were used to gather evidence regarding other online resources that the students used for learn testing and the type of information available on those resources. The students generally did not report any other online source. Regarding question 7, the results from a One-sample Wilcoxon Signed-Rank test showed that the average ratings of the "*benefit of using tools to support testing of programming assignments*" after using the WReSTT (at posttest) was significantly greater than 3 (i.e., midpoint of the scale) at p<0.05. Furthermore, a Paired-sample Wilcoxon Signed-Rank test revealed significant increase in the students' perception of benefits during the posttest when compared to their rating at the pretest (p < 0.01).

4.2 WReSTT Survey Results

After completing the posttest, the students were asked to complete a post-study survey that evaluated the student's overall response to the WReSTT, followed by more specific evaluation of the testing tutorials in WReSTT. The results of the students ratings (using a 5-point likert scale ranging from 1-Strongly Disagree to 5-Strongly Agree) is discussed in this subsection.



5. Do you know of any online resources that provide information on software testing?
Yes No

6. If you have answered "Yes" to Question (5), answer the following questions:

a. State the names of online resources.
WReSTT, Apple Developer, Wikipedia, YouTube, Udacity, among others

b. State the type of materials (notes, lab exercises, tutorials etc.) found at each resource listed above.
Tutorials, Video Tutorials, Tool Documentation, Quizzes, Notes on Testing, among others

7. How beneficial do you think it is to use tools to support testing of your programming assignments? Use a scale of 1 - 5 with *1 = Not beneficial* and *5 = Extremely beneficial*

Fig. 5 Rest of the questions

The first question in the survey collected a response (Yes/No) to determine what percentage of the students have ever used a learning source other than WReSTT to learn about testing concepts and tools. Based on the response from eighteen students, 83% (fifteen) of the students indicated no use of any other resource. This shows a lack of exposure of software testing in introductory programming course (in particular at our institution).

Regarding the feedback on the usability of the WReSTT, students rated the WReSTT on items 2 through 15 using a 5-point scale on different attributes (e.g., ease of use, ease of learn, expected functionality, clarity of information, recommendation) as shown in Table 1. The first two columns of Table 1 show the number and the description of the attribute being evaluated. The third column of Table 1 reports the mean and the standard deviation (S.D.) score from all the eighteen subjects for each attribute.

To evaluate the each attribute of WReSTT, we conducted a One-sample Wilcoxon Signed-Rank test to determine whether the mean ratings were significantly greater than 3 (i.e., midpoint of the scale). The test indicates that the WReSTT received significantly positive ratings on all fourteen attributes (i.e., p<0.05). The last column of Table 1 shows the p-value for each attribute.

Next, we evaluated the student's perception of the usefulness of the testing tutorials in WReSTT. For this purpose, students rated items 16 through 21 (as shown in Table 2) using a 5-point scale. Items 16-20 measured the student's response on usefulness of WReSTT tutorials in terms of overall quality, quantity, and with respect to its use on how to use the unit testing / code coverage / functional testing tools. The average score (S.D.) of each item is shown in Table 2. The result from a One-sample Wilcoxon Signed-Rank test that evaluated whether the mean ratings were significantly greater than 3 (i.e., midpoint of the scale) is shown in the last column of Table 2. The results showed that the WReSTT tutorials received significantly positive ratings of all the characteristics (p <0.05).

Table 1. Overall response to the WReSTT

#	Attribute	Mean (S.D.)	p-value
2	Overall, I am satisfied with how easy it is to use the website	3.92 (0.61)	<0.01
3	It is simple to use the website	3.78 (0.69)	<0.01
4	I feel comfortable using the website	3.5 (0.65)	<0.01
5	It was easy to learn to use the website	3.78 (0.57)	<0.01
6	I believe I became productive quickly using the website	3.35 (0.92)	<0.05
7	The information (such as online help, on page messages, and other documentation) provided with the web site is clear	4.07 (0.75)	<0.01
8	It is easy to find the information I need	3.78 (0.69)	<0.01
9	The information is effective in helping me complete the tasks and scenarios	3.61 (0.5)	<0.01
10	The interface of the website is pleasant	3.85 (0.77)	<0.01
11	I like using the interface of this website	3.64 (0.49)	<0.01
12	The website has all the functions and capabilities I expect it to have	3.78 (0.84)	<0.01
13	I believe that the website helped me earn a better grade	3.3 (0.65)	<0.05
14	I would recommend the website to fellow students	3.71 (0.82)	<0.01
15	Overall I am satisfied with the website	3.84 (0.64)	<0.01

Regarding the feedback collected from open ended questions, a significantly large number of students would like to introduce WReSTT in their future programming courses.

5. DISCUSSION OF RESULTS

The results from this study indicate the promise of using WReSTT to teach software testing in introductory computer programming courses. The students understanding of testing concepts, their knowledge of testing techniques, their tool usage and their perceived proficiency of the tool usage showed a significant improvement due to the exposure to the WReSTT. The results also show that the students had a significantly positive feedback of the WReSTT in terms of its usefulness, its ease and clarity of the information. Also, the results from the post-study survey showed that the WReSTT tutorials helped them understand the concepts and tools better and that they would not have used testing tools if they were not exposed to the WReSTT. An interesting result was the student's perception that the WReSTT helped them achieve a better grade in this class. We wanted to evaluate if this perception was true in reality by analyzing whether there was any positive correlation between the actual points scores on the programming assignments by the students (where WReSTT was used) vs. their rating of the belief that the website helped me earn a better grade. The result from a linear regression showed a significantly positive ($p= 0.028$, $r^2= 0.42$), that the students who had a significantly positive perception of WReSTT on a better grade achieved higher points on their programming assignments (where WReSTT was used).

6. CONCLUDING REMARKS

Based on these results, we would like to continue investigating the further use of WReSTT in other software engineering courses at NDSU and other institutions. We would also welcome collaborations with the researchers at other institutions to help generalize the study results. Since this was a first study of using WReSTT at NDSU, we did not used the “virtual points” concepts and some other social networking features (See Figure 1) in this study. We have already begin using these features in studies that are undergoing in the Fall 2013 semester at our institution. The results from these studies would help us provide better understanding of the impact of WReSTT's collaborative learning environment on the students learning of software testing concepts and tools over a course of the semester.

7. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant DUE-1225972. We would like to thank the students who participated in the study and the reviewers of the

Table 2. Response to the usefulness of testing tutorials in WReSTT

#	Attribute	Mean (S.D.)	p-value
16	The tutorials in WReSTT helped me to better understand testing concepts	3.84 (0.68)	<0.01
17	The tutorials in WReSTT helped me to better understand how to use unit testing tools	3.69 (0.75)	<0.01
18	The tutorials in WReSTT helped me to better understand how to use code coverage testing tools	3.41 (0.66)	<0.01
19	The tutorials in WReSTT helped me to better understand how to use functional testing tools	3.36 (0.8)	<0.05
20	The number of tutorials in WReSTT is adequate	3.61 (0.76)	<0.01
21	I would not have used testing tools in my project if WReSTT did not exist.	3.92 (0.75)	<0.001

study design and the earlier version of the paper

8. REFERENCES

[1] Timothy C. Lethbridge. 2000a. Priorities for the education and training of software engineers. J. Syst. Softw 53, 1 (July 2000), 53–71. http://dx.doi.org/10.1016/S0164-1212

[2] Andrew Begel and Beth Simon. 2008a. Novice software developers, all over again. In Proceedings of the Fourth international Workshop on Computing Education Research (ICER '08). ACM, New York, NY, USA, 3–14. http://dx.doi.org/10.1145/1404520.1404522 .

[3] Jeffrey C. Carver and Nicholas A. Kraft. 2011. Evaluating the testing ability of senior-level computer science students. In Software Engineering Education and Training (CSEE T), 2011 24th IEEE-CS Conference on. 169–178. http://dx.doi.org/10.1109/CSEET.2011.5876084

[4] Vahid Garousi and Tan Varma. 2010. A replicated survey of software testing practices in the Canadian province of Alberta: What has changed from 2004 to 2009? J. Syst. Softw. 83, 11 (Nov. 2010), 2251–2262. http://dx.doi.org/10.1016/j.jss.2010.07.012

[5] WReSTT Team. Web-based Repository for Software Testing Tools, 2010. http://wrestt.cis.fiu.edu/ .

[6] ACM/IEEE-CS INTERIM REVIEW TASK FORCE. 2008. Computer Science Curriculum 2008: An Interim Revision of CS 2001. http://www.acm.org/education/curricula/ComputerScience2008.pdf

[7] ASTIGARRAGA, T., DOW, E., LARA, C., PREWITT, R., AND WARD, M. 2010. The emerging role of software testing in curricula. In Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments, 2010 IEEE. IEEE, Piscataway, NJ, USA, 1–26

[8] T. C. Lethbridge, J. Diaz-Herrera, R. J. J. LeBlanc, and J. B. Thompson. Improving software practice through education: Challenges and future trends. In FOSE '07: 2007 Future of Software Engineering, pages 12–28, Washington, DC, USA, 2007. IEEE Computer Society.

[9] P. J. Clarke, A. A. Allen, T. M. King, E. L. Jones, and P. Natesan. Using a web-based repository to integrate testing tools into programming courses. In Proceedings of the ACM OOPSLA 2010 Companion, SPLASH '10, pages 193–200, New York, NY, USA, 2010. ACM.

[10] P. J. Clarke, J. Pava, Y. Wu, and T. M. King. Collaborative web-based learning of testing tools in se courses. In Proceedings of the 42nd SIGCSE Conference, pages 147–152, New York, NY, USA, 2011. ACM

Appendix A

PRETEST/POSTTEST

This test will NOT impact your course grade.

ID Number (DO NOT use your name):

1. What is the main objective of program testing?

2.(a) Identify all the testing techniques you have used to create test cases:

(b) Write test cases to test the code provided below. Identify the testing technique used to generate the test case.

```
//Method to withdraw money from a bank account
//The following variables are defined as follows:
//requiredMin = 50.0 and balance = 100;
 public void withdraw (double amount) {
   if ((balance - amount) < requiredMin)
    System.out.println("Insufficient funds");
   else
    balance = balance - amount;}
```

Input Value	Expected Output Values			Testing Technique
amount	amount	balance	requiredMin	

3. Have you ever used tools to support testing of programs? Circle your answer: Yes No

4. If you have answered "Yes" to Question (3), answer the following questions:

a. List the names of the tools you have used:

b. List one tool in each of the following categories that you have used and indicate your level of proficiency corresponding to each tool on a scale of 1 - 5 with, *1 = barely competent* and *5 = extremely proficient*:

	Category	*Tool(s)*	*Proficiency*
i	Unit Testing		
ii	Functional Testing		
iii	Code Coverage		

5. Do you know of any online resources that provide information on software testing? Yes No

6. If you have answered "Yes" to Question (5), answer the following questions:

a. State the names of online resources.

b. State the type of materials (notes, lab exercises, tutorials etc.) found at each resource listed above.

7. How beneficial do you think it is to use tools to support testing of your programming assignments? Use a scale of 1 - 5 with *1 = Not beneficial* and *5 = Extremely beneficial*

8. If you answered 2 or above to Question (7), state one reason for your answer.

.

THE EFFECT OF CHANGING PROGRAMMING LANGUAGE TO STUDENT SUCCESS IN UNDERGRADUATE COMPUTER SCIENCE CURRICULUM

Ebru Celikel Cankaya
University of Texas at Dallas, Richardson, TX, USA
{ebru.cankaya@utdallas.edu}

Abstract- . We measure and compare the effects of changing the programming language in introductory level courses (Computer Science I and II) in undergraduate Computer Science (CS) curriculum. The study presents the example of University of Texas at Dallas (UTD) case, where we present the results of two approaches: Approach 1 where both CS I and CS II courses are taught in Java, and Aprroach 2 where CS I is taught in C++, and CS II in Java. The paper presents the data collected from four consecutive semesters and analyzes and compares the effect of using different programming languages on student success, as well as drop rates. The results show that changing the programming language in consecutive sequences of introductory level undergraduate courses in CS curriculum do not adversely affect the student performance, and in some presented cases it even helps students score better. Also, we find that this change causes once an improvement, and another time a deterioration in the drop rate, which suggests the requirement for more data to be collected in coming semesters.

Key Words- Programming language, CS I, CS II, C++, Java, undergraduate curriculum.

1. INTRODUCTION

As any education in a specific field of science would, Computer Science education also takes enthusiasm, dedication, an ability to have analytical thinking skills and to use them properly. Therefore, it gets utterly important to attract and retain students to this demanding major in colleges and universities. There are factors that obstruct us from achieving this goal: Ironically, we need to employ proven to have worked methods as otherwise will be too costly. In the meantime, we need to be open to trying new methods so as to see if they work for us. Moreover, we need to repeat trying a new method several semesters to observe consistency and to determine if the result (either appearing to be an improvement or decline). There is a tradeoff behind all these as we can hardly recover from any losses that took place.

This study focuses on the effect of changing the programming language in introductory level courses at undergraduate Computer Science curriculum to the success of students, and therefore to the retention rate. It presents the results of a 6 semesters long study obtained at introductory level programming courses at undergraduate level at University of Texas at Dallas (UTD) Department of Computer Science. Specifically, we measure the effect of teaching CS I and CSII courses as once both in Java, then as CS I in C++ and CS II in Java. We compare the outcomes based on the students' success rates per assessment and see that changing the programming language has no significant effect on the performance of students that is measured in numeric grade. Furthermore, the results also show that the change in the programming language had no significant effect on the overall retention rate for these students.

The rest of this paper is organized as follows: In section 2, we present similar efforts that are adopted in higher education institutes nationwide and even internationally that aim at improving Computer Science education in undergraduate level as well as keeping the retention rates high. Section 3 shows in detail the outcomes we obtained from the data gathered from totally four consecutive semesters at UTD Computer Science Department. Section 4 concludes the work and presents some insight on future work to expand the current study.

2. BACKGROUND AND MOTIVATION

The effect of changing the programming language in introductory level courses at undergraduate Computer Science curriculum has direct consequences as changes in student success, and the retention rate. To improve retention among CS students, several methods have been introduced and are being applied. As mentioned in literature: Some of these efforts focus on a particular group such as minorities, women (Haines, 2013), needy students, etc. Among these methods lie such activities as organizing computer related workshops for prospective students at high school - or even at middle school - level so that they can play, program, and enjoy using computers and hopefully develop interest towards computer science (Robinson, Pérez-Quiñones, 2014, Yardi 2007), or taking freshmen CS major students to targeted conferences, such as the Grace Hopper Celebration of Women in Computing (Alvarado, Judson, 2014), or providing extra help to students via mentoring programs (Brown, Yuan, 2014), incorporating a peer led team learning (PLTL) scheme that involves participating students into small groups (4-8 students) to meet regularly for additional problem solving sessions in the lead of a peer leader (Horwitz et al., 2009). Other scholar work focuses on developing social bonds among CS majors to nourish and reinforce community identity (Crenshaw et al., 2008). Yet another approach is to change the curriculum, such as incorporating a theme based approach and supporting hands-on labs (Barr, 2012).

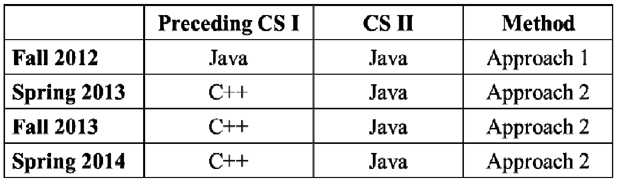
Some of these methods are general enough to get better retention rates in almost any major in Universities. There are also more specific actions we can take such as choosing a programming language that will engage and keep student interest in programming. Newhall et al. 2014 presents the successful results they have obtained by changing the programming language from several years in C, then one year in Java, and afterwards adopting Python in Swarthmore College. Similarly, we present the effect of programming language on student success, though we specifically analyze changing programming language in the two introductory level CS courses: CS I and CS II at UTD.

All of the above mentioned methods have been applied successfully and yielded promising results in improving student retention rates in undergraduate, as well as graduate levels. In this work, in addition to the eventual goal of improving retention rates, we also focus purely on student success: We investigate if exposing students to two different programming languages, namely C++ and Java, at introductory level programming sequence undergraduate courses CS I and CS II, or covering both courses (CS I and CS II) in one single programming language (in Java) yields better performance results. For this, we use student score as a measure of performance.

3. EXPERIMENTAL WORK AND RESULTS

We base our experimental results on the data collected on four consecutive semesters at two undergraduate introductory level CS major courses as CS I and CS II at University of Texas at Dallas (UTD) Department of Computer Science.

Table 1: The programming languages used for each CS I and CS II courses during four consecutive semesters.



	Preceding CS I	CS II	Method
Fall 2012	Java	Java	Approach 1
Spring 2013	C++	Java	Approach 2
Fall 2013	C++	Java	Approach 2
Spring 2014	C++	Java	Approach 2

As seen in Table 1, at first Java was used as the programming language for teaching both CS I and CS II introductory level undergraduate courses in Fall 2012. Then, starting with Spring 2013, during four consecutive semesters, where Preceding CS I column indicates the CS I level course that students needed to take as a prerequisite for CS II. For the sake of ease of reference, we define and refer to the following labels for two separate approaches: Approach 1: Both CS I and CS II are taught in Java (only in Fall 2012). Approach 2: CS I is taught in C++ and CS II is taught in Java (starting with Spring 2013, continuously through Spring 2014).

Deciding which programming language to use is a Departmental level decision and the reason behind choosing Java (and C++) is to help students get exposed to an object oriented programming language so as to learn fundamental CS concepts easier and in a popular way (rather than conventional procedural languages).

It should also be noted that the main focus of CS I and CS II courses is to teach fundamental CS concepts and a programming language is used as a means to help program sample code while teaching these topics. According to UTD Department of Computer Science course curriculum (UTD Course curriculum URL, 2014), the CS I course introduces fundamental concepts as introduction to object-oriented software analysis, design, and development, classes and objects, object composition and polymorphism, sorting, searching, recursion, strings using core classes, inheritance and interfaces, and Graphical User Interfaces. The CS II course teaches more advanced topics as exceptions and number formatting, file I/O, implementation of primitive data structures - including linked lists, stacks, queues, and binary trees -, advanced data manipulation using core classes, and introduction to multi-threading, multimedia, and networking.

To examine how students perform under each approach, i.e. Approach I and Approach II, we measure student performance (based on numeric grade out of 100) per semester. Figure 1 illustrates these performance values.

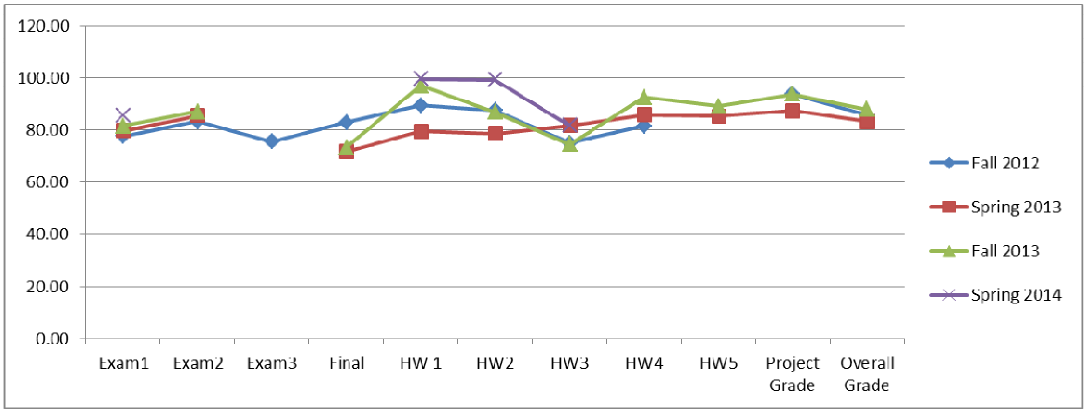


Figure 1. Plot for average student performance per assessment for each semester.

According to Figure 1, student success rates under Approach 1 (in Fall 2012, when both CS I and CS II are taught in Java) outperform Approach 2 (in Spring 2013 through Spring 2014, when CS I is taught in C++ and CS II in Java) only in the final exam assessment type. For all other assessments, students either perform the lowest under Approach 1, or their grades are better at least in one of the Approach 2 based semesters. It should be noted that in Figure 1, some assessment scores are not applicable to all semesters, (such as there is no Exam 3 in Spring 2013 and Fall 2013, and for the Spring 2014 semester, some grades are not available yet as the semester still continues as of March 2014).

Table 2: Average student performance data per assessment for each semester.

Semester	# of students	Exam1	Exam2	Exam3	Final	HW 1	HW2	HW3	HW4	HW5	Project	Overall Grade
Fall 2012	185	77.52	83.28	75.39	82.82	89.37	87.51	74.95	81.44		93.65	85.53
Spring 2013	45	79.53	85.44		71.71	79.41	78.52	81.57	85.72	85.30	87.26	83.11
Fall 2013	120	81.42	86.93		73.19	96.86	86.75	74.36	92.50	89.02	93.65	87.78
Spring 2014	47	85.55				99.52	99.11	81.60				

Table 2 lists the numeric scores that Figure 1 bases itself on. We solely focus on student performance in CS II, as this should shed light on whether adopting Approach 1 or Approach 2 helps students perform better. Therefore, the *# of students* column in Table 2 contains the number of students enrolled to CS II course for the corresponding semester. Table 2 shows that Exam 1 scores continue to improve as we progress chronologically through semesters. Exam 2 scores are very close to each other, though with insignificant differences. Students seem to have performed the best in the Final exam, when UTD adopted Approach 1, and it deteriorated later on. In terms of homeworks, students seem to be better doing in each homeworks. Actually, the underlying reasons behind them should not only be related to solely student effort, or to the case where Approach 2 is adopted, but also the instructors should be credited for putting their experience and insight into making it a smoother transition to switch from C++ to Java. Still, it would be too soon to conclude that all semesters under Approach 2 do better on average, as some data are not available by the end of Spring 2014.

In order to analyze individual student performances per semester, we plot the charts in Figures 2, 3, and 4 for Fall 2012, Spring 2013, and Fall 2013 respectively. For privacy concerns, student names are masked with alphabet letters.

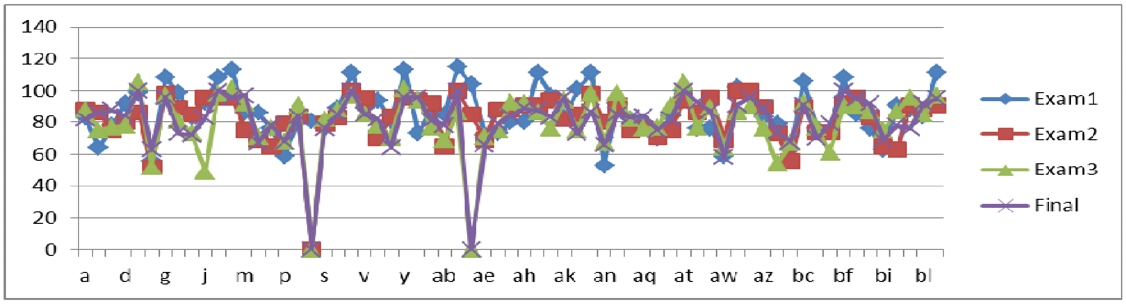


Figure 2. Individual student performances per assessment in Fall 2012.

As seen in Figure 2, individual students perform consistently good (or bad) in the 3 midterm exams and Final exam for Fall 2012. It should be noted that for visibility purposes, the chart in Figure 2 is plot on selected data (particular students and particular assessments only, even though the full set of data is available for all 185 students and 11 assessment categories). The sudden drops in the figure shows students who did not participate in this particular assessment.

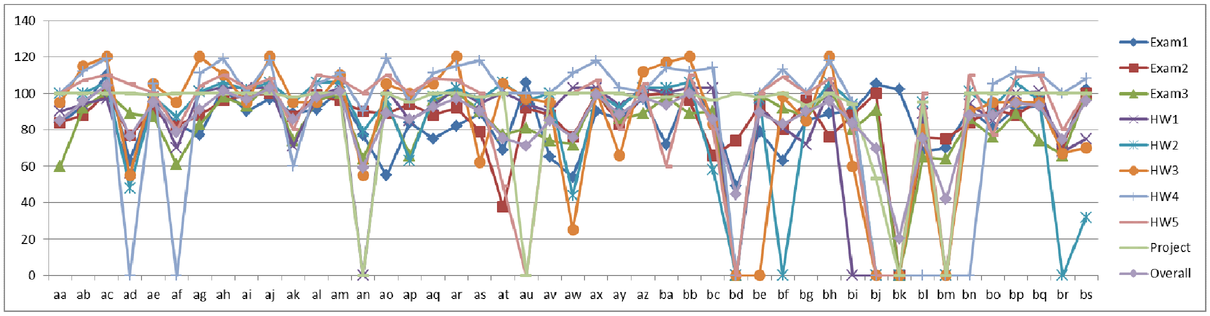


Figure 3. Individual student performances per assessment in Spring 2013.

Figure 3 illustrates the student performance for 2 midterm exams, Final exam, 5 hws and project and overall grade on an individual basis for Spring 2013. According to the figure, individual student performances are consistent per assessment.

Similar to Figures 2 and 3 above, Figure 4 shows the individual student performances on arbitrarily selected students and assessment categories demonstrate a consistent trend.

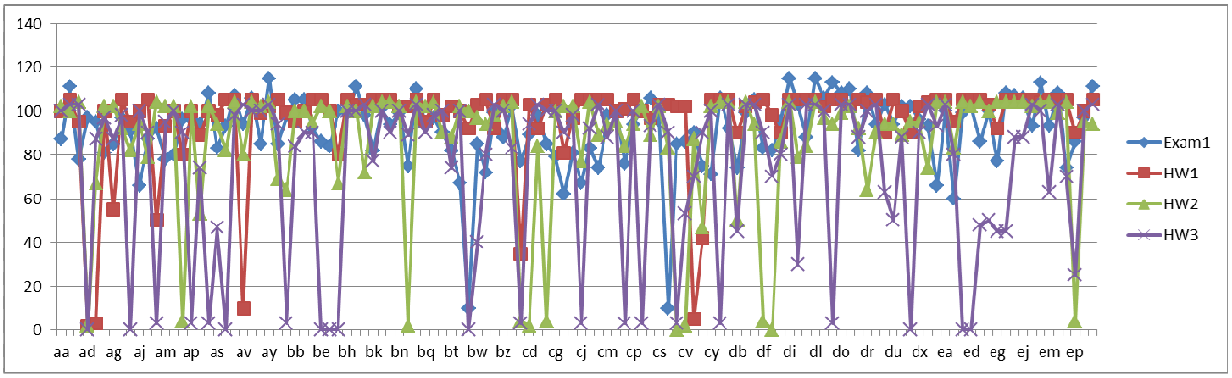
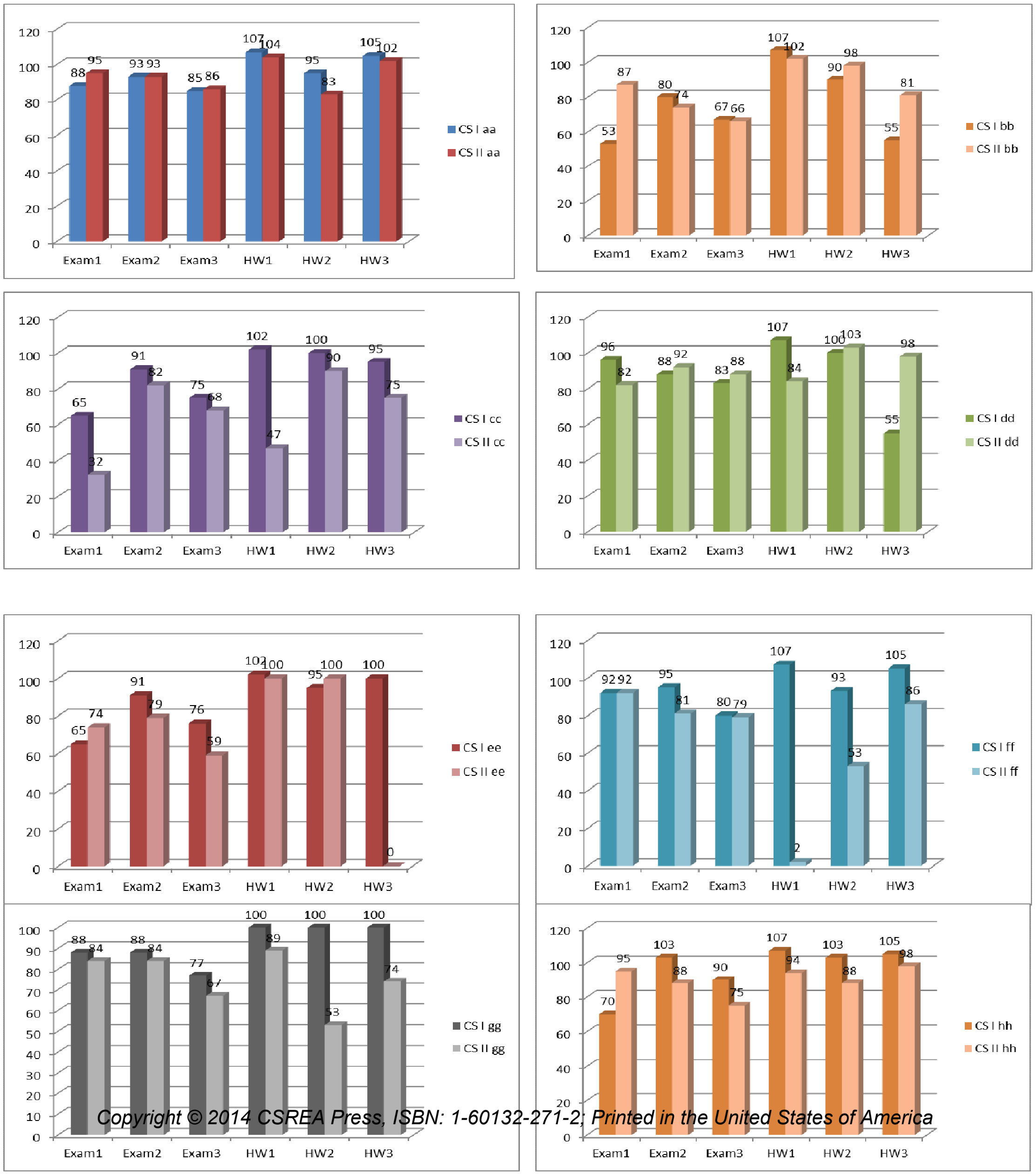


Figure 4. Individual student performances per assessment in Fall 2013.

Our work also explores the effect of incorporating Approach 2, i.e. how students perform when they take CS I in C++, and CS II in Java. There are 10 such students monitored, and we present the performance scores for these students who took CS I in Spring 2013, and CS II in Fall 2013. Again, student names are masked with alphabet letters for privacy (Figure 5).



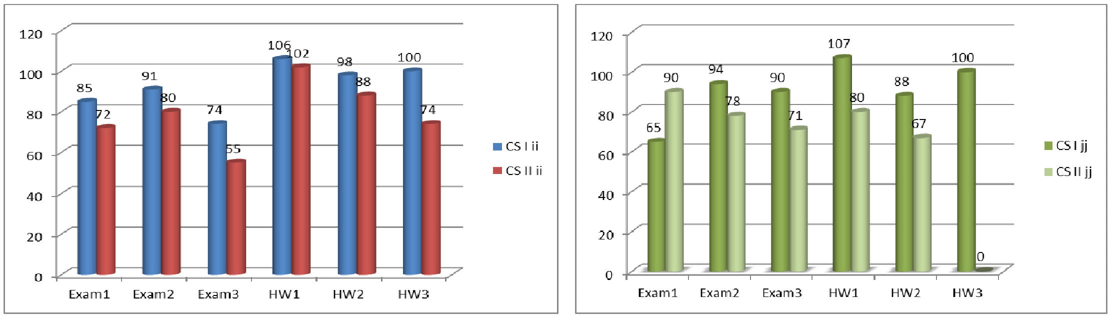


Figure 5. Same student taking CS I in C++ (Spring 2013) and CS II in Java (Fall 2013).

According to Figure 5, in CS II (in Java) the same student performed sometimes better, and some other times worse than how he performed in CS I (in C++). Specifically, for Exam 1: on average in 6 out of 10 cases, students perform better (or the same) in CS II with Java, when they had CS I in C++. This actually explains why Exam 1 performs the poorest in Fall 2012, when Approach 1 (both CS I and CS II in Java) was employed (Table 1). For Exam 2 and Exam3: Only in 20% of Exam2 (as well as Exam3) grades are equal or better in CS II (in Java), which follows CS I taught in C++. For HW1: All students get higher scores in CS I (in C++) then they do in CS II (in Java). This explains well that students need to gain some experience and acquaintance to start performing satisfactorily in a new programming language. Starting with HW2, 30% of students start to score better in CS II with Java, in spite of a C++ background in CS I. And finally for HW3, 20% of the students could achieve satisfactory (better than or equal) scores with CS I (in C++) and CS II (in Java). For HW3, it should be taken into consideration that some students have totally missed the submission and this is why they scored that low.

One of the goals of this paper is to observe the effect of changing the programming language in two introductory level courses into retention rates. To investigate this, we inquire the course drop rates for the four semesters that we have collected the statistics for. Certainly, our focus is on CS II, as it is the course where we see the immediate results of changing the programming language from what was taught in CS I (that is C++) to Java. For the sake of generality, we calculate the *Failures* column as the total of number of F (failure) grades and W (withdraw) end of semester grades for the CS II course per semester. Table 2 shows these values.

Table 2. Drop rates per semester.

	Method	**# of students**	**Failures**	**Drop rate**
Fall 2012	Approach 1	185	6	3.24%
Spring 2013	Approach 2	45	4	8.89%
Fall 2013	Approach 2	120	2	1.67%
Spring 2014	Approach 2	42	N/A	

We see that drop rates under each approach vary in a large span. Though in Fall 2012 under Approach 1 the drop rate is quite low at 3.24%, in Fall 2013 under Approach 2 this rate is even lower at 1.67%. So, this can be considered as a promising result. Still, the considerable jump at 8.89% drop rate is a significant change that needs to be investigated to make sure whether it is an isolated result, or an ongoing tendency with data in coming semesters.

4. CONCLUSION AND FUTURE WORK

We present the four consecutive semester results of changing programming language in two introductory level courses CS I and CS II in Computer Science major at University of Texas at Dallas (UTD) Department of Computer Science. We first analyze Approach 1 where both CS I and CS II are taught in Java, then Approach 2, where CS I is taught in C++ and CS II in Java. Results show that as opposed to the prejudice, students perform better (or at least now worse) under Approach 2, as compared to Approach 1.

We also investigate if this change causes increased drop rates. Our results show that for one semester Approach 2 yielded better (lower) drop rates than Approach 1, and in another worse (higher) rates than Approach 1. This indicates that the study needs to be expanded to future semesters to reach a solid conclusion.

As part of the future work, authors are considering to involve other factors, such as course content, measuring the effect of mentoring efforts into the success rates of students, as well as the drop rates.

5. REFERENCES

Alvarado C., Judson E., “Using Targeted Conferences to Recruit Women into Computer Science”, Communications of the ACM, Vol. 57, Issue 3, March 2014, pp. 70-77.

Barr V., “Create two, three, many courses: An experiment in contextualized introductory computer science”, Journal of Computing Sciences in Colleges , 27(6), pp. 19-25.

Brown S., Yuan X., “Techniques for retaining low performing students: high-need student mentoring program”, SIGCSE’14, pp. 708.

Crenshaw T. L., Chambers E. W., Metcalf H., Thakkar U., “A Case Study of Retention Practices at the University of Illinois at Urbana-Champaign”, SIGCSE’08, March 12–15, 2008, Portland, Oregon, USA, pp. 412-416.

Haines A., “How one college president is breaking down barriers for women in tech”, Forbes;http://www.forbes.com/sites/85broads/2011/12/12/how-one-college-president-is-breaking-down-barriersfor-women-in-tech/ [Accessed on 3/11/2014]

Horwitz, S., Rodger, S. H., Biggers, M., Binkley, D., Frantz, C. K., Gundermann, D., Hambrusch, S., Huss-Lederman, S., Munson, E., Ryder, B., and Sweat, M. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science”, SIGCSE 2009, pp. 163-167.

Newhall T., Meeden L., Danner A., Soni A., Ruiz F., Wicentowski R., “A Support Program for Introductory CS Courses that Improves Student Performance and Retains Students from Underrepresented Groups”, SIGCSE’14, March 5–8, 2014, Atlanta, GA.

Robinson A., Pérez-Quiñones M. A., “Underrepresented Middle School Girls: On the Path to Computer Science through Paper Prototyping” SIGCSE’14, March 5–8, 2014, Atlanta, GA, USA.

UTD Course curriculum URL, 2014: http://coursebook.utdallas.edu/

Yardi S., Bruckman A., “What is computing? bridging the gap between teenagers' perceptions and graduate students' experiences. ICER '07 International Computing Education Research Workshop, Atlanta, GA, September 15 - 16, 2007, pp. 39-49.

Using Visual Logic with Pseudocode to Teach an Introductory Programming Course

G. Cooper, PhD
Engineering and Information Sciences, Devry University, Downers Grove, IL, USA

Abstract - *Introductory programming courses experience large attrition rates due to the challenges of the material and incoming student learning styles. CIS115 is an introductory logic and design course at DeVry University. To improve student satisfaction and reduce attrition in the course, Visual Logic (www.visuallogic.org) with pseudocode was introduced as an alternative to a standard programming language. Using Visual Logic, students were able to create executable flowcharts while learning fundamental programming concepts. Pseudocode was taught in conjunction with Visual Logic so that students learned to translate the flowchart into workable code. Visual Logic was employed in both an online and onsite delivery of the course. Student satisfaction survey data and grades were evaluated. Grades from this course were compared with the course when a high level programming language was taught. This paper explains the outcomes of using Visual Logic with pseudocode in an introductory programming course and explores the advantages and disadvantages of this approach.*

Keywords: Visual Programming, Programming concepts, Introductory Course, Logic and Design, Computer Science

1 Introduction

Introductory programming courses are faced with several challenges. Students enter the course with varying degrees of prior knowledge and comfort with programming. CIS115 is a Logic and Design course at DeVry University. The course introduces program design and basic logic with pseudocode and flowcharts. No programming languages are used in the course; rather all assignments use Visual Logic and pseudocode.

Visual Logic is an executable flowchart program utilizing a graphical interface to illustrate programming topics. The flowchart provides users with the ability to create loops, decisions, arrays, procedures, and files. Users execute the flowchart to produce results and can interact with the flowchart using basic input/output commands. Visual Logic removes the frustration of syntax errors and new integrated development environments (IDEs) from novice programmers. The challenge is in creating the logic rather than managing a new development environment.

Students use Visual Logic in the first programming course they take. The course is delivered in both an online and onsite format with the software available on a remote Citrix site. Students login to the site, create flowcharts to implement their assignments and store the Visual Logic files on the remote site.

Research has shown that students can learn to program effectively using graphical tools and non-standard programming environments [6][7][8]. Multiple graphical-based program teaching tools have been developed and implemented at universities. Most notably, Alice, developed by Carnegie Mellon in 1999 [3] provides new programmers with an object oriented approach to programming. Alice offers students the ability to create 3D animations while learning fundamental programming topics. RAPTOR, created by the Air Force Academy in 2005 [2] uses a flow model environment to teach programming concepts. In a study of three final exam questions given to students with the option of using traditional programming languages or RAPTOR, students choosing RAPTOR to complete the final exam questions scored significantly higher than those using traditional programming languages [2].

In addition to these results, in 2011 Gudmendsen et al. completed a study using Visual Logic in three different approaches to determine its effectiveness in the classroom[5]. The first course used Visual Logic to introduce algorithms in a general education computer science course; the second course used Visual Logic for six weeks then transitioned to Python. The third course used Visual Logic in conjunction with teaching Java. In all three courses using Visual Logic enhanced student learning.

This paper takes a new approach by introducing Visual Logic in conjunction with pseudocode. The pseudocode is presented as equal in importance to Visual Logic in learning logic. To determine the effectiveness of Visual Logic with pseudocode in CIS115, two areas were examined. End of course student surveys and grades for four sessions of CIS115 was gathered. This data was compared with four sessions of the course when a high level programming language was taught. The question this research sought to answer was "do more students understand the material, enjoy the course, and pass the course when Visual Logic is taught as compared to a high level programming language?" This paper describes the findings and extrapolates on methods to improve the course.

2 Initial Course Design

The Logic and Design course is one of the first courses that new freshmen take in their major. This course is taught in a lab/lecture classroom in which each student has a computer at his or her desk. In addition to this format, the course is also offered online. CIS115 is a 3 credit hour class and the course description is below:

This course introduces basic programming logic, as well as algorithm design and development, including constants, variables, expressions, arrays, files and control structures for sequential, iterative and decision processing. Students learn to design and document program specifications using tools such as flowcharts, structure charts and pseudocode. Program specification validation through desk-checking and walk-throughs is also covered.

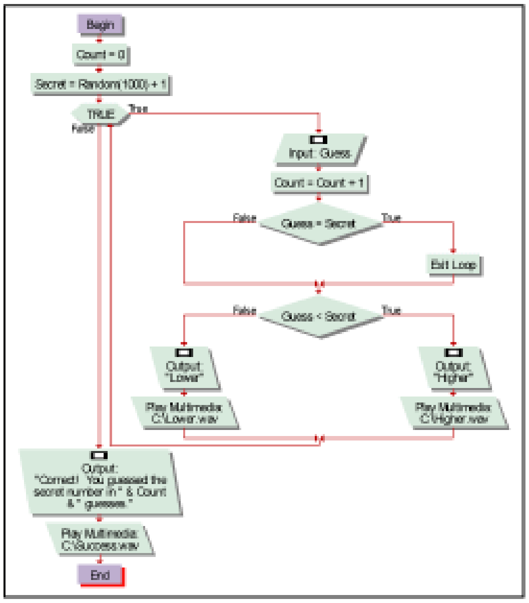
To assist students with the transition from logic and design to programming, pseudocode is an important part of this course. Students must create pseudocode that matches the Visual Logic program. Prior to introducing Visual Logic, new programming students used C++. The same topics were covered in the course regardless of the teaching tool.Length

The maximum allowed number of pages is seven for Regular Research Papers (RRP) and Regular Research Reports (RRR); four for Short Research Papers (SRP); and two for Posters (PST).

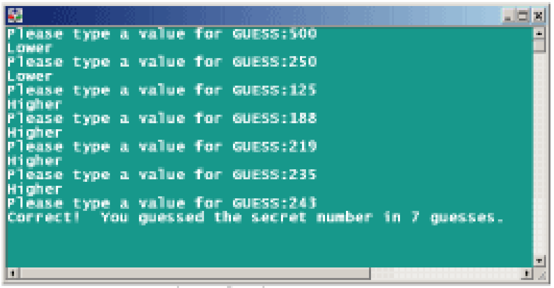
3 Visual Logic

Visual Logic provides students with a visual method to create programs. All assignments require students to create an IPO (input, processing, output) chart, a Visual Logic flowchart, and pseudocode.

Visual Logic (visuallogic.org) is a visual flowchart program available for download as a trial version [9]. The program can be purchased by universities or individuals. Visual Logic will only run on Windows operating systems, which presents a problem for students using other operating systems. The university has a centralized server for students to access using a Citrix client. Visual Logic is available for all students with a login to the site. Figure 1 shows an example of a Visual Logic screenshot.


Begin
Count = 0
Secret = Random(1000) + 1
TRUE
True
False
Input: Guess
Count = Count + 1
False
Guess = Secret
True
Exit Loop
False
Guess < Secret
True
Output: "Lower"
Output: "Higher"
Play Multimedia: C:\Lower.wav
Play Multimedia: C:\Higher.wav
Output: "Correct! You guessed the secret number in " & Count & " guesses."
Play Multimedia: C:\Success.wav
End

Visual Logic Flowchart solution to the High-Low Game



```
Please type a value for GUESS:500
Lower
Please type a value for GUESS:250
Lower
Please type a value for GUESS:125
Higher
Please type a value for GUESS:188
Higher
Please type a value for GUESS:219
Higher
Please type a value for GUESS:235
Higher
Please type a value for GUESS:243
Correct!  You guessed the secret number in 7 guesses.
```

Visual Logic flowchart execution output

Figure 1: Visual Logic flowchart and output

Using Visual Logic, students can create complex programs with multiple programming structures. Visual Logic has no specific data types so data types are implemented in the pseudocode. Even so, Visual Logic supports arrays, if statements, pre and post-test loops, text file I/O, logical operators, and procedures.

Students create complex flowcharts with Visual Logic and learn basic logic problem solving. They can run their flowcharts to determine the accuracy of their design and give themselves a sense of accomplishment.

4 Visual Logic with Psuedocode Implementation

Visual Logic provides the opportunity for students to witness their programs running without the usual syntax errors that come from learning a new programming language. Each assignment is composed of an IPO chart, pseudocode, and a Visual Logic flowchart. The IPO chart is a high level overview of the problem the students are solving. For example, if the problem is to calculate the area of a circle after the user enters the radius, the input would be the radius, output is the area of the circle and the processing is area = 3.14 * radius * radius.

The students write pseudocode with basic syntax. The students use keywords such as: START, DECLARE, PROMPT, INPUT, DISPLAY, CALCULATE, SET, IF/THEN/ELSE, DO/WHILE, FOR, WHILE, SWITCH, STOP. Since pseudocode is language-independent, only three data types are discussed in the class. These data types are real, integer, and string.

The following is an example of pseudocode to calculate the area of a circle after the user inputs the radius.

```
START
  DECLARE Real radius.
  DISPLAY "Enter radius: "
  INPUT radius
  DECLARE Real AreaOfCircle
  SET AreaOfCircle = 3.14159 * radius * radius
  DISPLAY "The area is: "+AreaOfCircle
STOP
```

Relational operators and Boolean logic are covered with selection and repetition structures. The pseudocode is stressed to the students as one of the most important parts of the course. Pseudocode translates nicely to high level programming languages.

The assignments account for 52% of the total course grade. The remaining points are distributed between threaded discussions, quizzes, and a comprehensive final exam. At the end of the course students have a firm foundation in logic and design and are well prepared to start programming in a high level language.

Visual Logic has several advantages for beginning programmers. The main benefit is the visual nature of the program. Students can visualize input, output, processing, decisions, and repetition in their flowchart. They can create large flowcharts with relative ease and run their programs to see them working. Most students are comfortable working with Visual Logic after one short tutorial. Another advantage of Visual Logic is that it can be used with only a web browser and internet connection. Students login to the Citrix client and use Visual Logic from there. Some students choose to download the trial version, but most access the program using Citrix. Since CIS115 is taught in both the online and onsite modality the ease with which the students can access the software is an important factor.

Writing pseudocode using a word processor has the advantage that students can focus on logic instead of coding syntax. Many IDEs have their own learning curves whereas pseudocode can be written using a word processor.

While Visual Logic has many benefits over traditional programming to introduce new programmers to logic and design, students fear they will not use Visual Logic again. They are eager to start programming and use their new skills with a high level programming language. Visual Logic is basically a teaching tool and they may not encounter this program in industry.

5 Results and Discussion

Two areas of student achievement were studied to determine the success of Visual Logic with pseudocode in CIS115. These areas were end of course student surveys and grades. Pass rates in introductory programming courses are notably low [1]. Bennedsen and Caspersen studied institutions around the world and the pass rate for a computer science 1 course. The average pass rate in these courses is 67%.

Each session, online surveys are administered to students. These surveys are optional, yet students are encouraged to complete them to give the instructor feedback on the course. The survey consists of six main areas of feedback and a comment section. The composite score of the course along with the comments were evaluated. Students rated the course from 1 to 10 with 10 being the highest score.

The mean student satisfaction with the course was 8.57 for all students. For the onsite course the mean satisfaction was 9.75 and the online student satisfaction was 8.42. Four onsite students participated in the survey and 32 online students participated. It is interesting to note that onsite students had approximately 13.7% higher satisfaction with the course than online students. Table 1 illustrates the number of students who responded with scores from 1 to 10.

These results are noteworthy since student satisfaction can lead to more motivated learners, higher grades, and better persistence. Over half of the students rated CIS115 with a 10, the highest score and 72% of students rated CIS115 with a 9 or 10. This indicates a high degree of satisfaction with the course overall. This student satisfaction survey was not administered to students when the course was delivered with C++.

Table 1. Student Satisfaction Results based on Survey Data

Course Score	**Number of student respondents**
10	20
9	6
8	4
7	1
6	1
5	2
4	2
3	0
2	0
1	0

In addition to student satisfaction surveys, student grades were assessed. The number of students who achieved grades of A-C, and those who passed and failed were tabulated for four sessions of CIS115 and are illustrated below in Table 2. This data does not include students who dropped the course, only those who completed the course and took the final exam.

Table 2: Student Grades for Four Sessions of CIS115

	Percent Grades A-C	**Percent grade D**	**Percent Passed (grades A-D)**	**Percent Failed**
Before Visual Logic	78.57%	12.5%	91.07%	8.93%
After Visual Logic	76.93%	10.68%	87.38%	12.62%

While the pass rates of both implementations of the course are high, more students passed using a high level programming language. The pass rate for C++ is approximately 4% higher than the pass rate for Visual Logic. While the students enjoy CIS115 and about 3 out of 4 students had a reasonable understanding of the subject material, more students passed the course when C++ was used rather than Visual Logic. While Visual Logic was intended to improve the pass rate, it actually decreased the pass rate slightly.

The results were surprising but can be explained when further examined. One of the main reasons to use Visual Logic is to reduce frustration from learning syntax and IDEs. Microsoft Visual Studio, the IDE used when programming C++ by Devry University students is user friendly as compared to other IDEs or using text editors.

Another advantage of using C++ is that students will use this language in their careers whereas Visual Logic is basically a training tool. Many students complained that they would never use this program again. Students who start immediately in a programming language can work through problems in an actual IDE and test their algorithms with code. Students may be more motivated to learn the material when they are learning a programming language that they can expect to use in their careers as opposed to Visual Logic which they do not expect to ever see outside of an academic environment. They can find satisfaction in using an actual language and creating programs in that language.

6 Course Improvement

The results from this study indicate that while students are satisfied using Visual Logic with pseudocode in their introductory programming course, they are not as successful as using a high level programming language. These results are noteworthy, and are counter to the results found from other researchers [2][5]. This course can still be improved to increase student comprehension and reduce frustration with syntax errors.

Writing pseudocode prior creating programs is a good habit for the students to learn. Instead of combining pseudocode with Visual Logic, pseudocode could be combined with C++ or Visual Basic. Visual Basic is a user friendly programming language with relatively easy syntax to understand.

Another improvement to the course would be to introduce a scripting language to coincide with Visual Logic and the pseudocode. A scripting language such as python would be easier to learn and a good marketable skill for the students.

An area of further study would be determining the extent to which the skills learned in CIS115 transfer to the subsequent course. Student survey data and grades from the next course could be analyzed to ascertain their satisfaction with the course and mastery of the subject material.

These options for course improvement provide opportunities for students to test the knowledge they gained from CIS115. A combination of Visual Logic, pseudocode, and a scripting language or Visual Basic and pseudocode can be implemented and the impact on student satisfaction, retention, and grades analyzed.

7 Conclusions

Introductory computer programming courses are challenging courses for students to take and instructors to teach. Determining the best method of explaining logic and design to new programmers while balancing the aggravation from syntax errors can be difficult. To improve the student experience Visual Logic with pseudocode was introduced in CIS115. The course was previously taught using C++. Data was analyzed for CIS115 to determine student satisfaction, knowledge of the subject material, and pass rates. The data consisted of student satisfaction from end of course surveys, and grades. The course had an 87% pass rate with 72% of students rating the course with a 9 or 10 out of 10. However, prior to using Visual Logic the course had a 91% pass rate indicating that using Visual Logic with pseudocode had little effect on improving student retention and understanding of the material.

To challenge students and improve their comprehension of the material, adding a scripting language or Visual Basic were explored and explained in the Course Improvement section. Based on the findings of this analysis, using Visual Logic with pseudocode to introduce logic and design was not as successful as using C++ for CIS115 at DeVry University.

8 References

[1] Bennedsen, J. and Caspersen, M. "Failure Rates in Introductory Programming." ACM SIGCSE Bulletin, v. 39 n.2 2007.

[2] Carlisle, M., Wilson, T., Humphries, J. and Hadfield, S. "RAPTOR: A Visual Programming Environment for Teaching Algorithm Problem Solving." SIGCSE '05 Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. p 176 – 180.

[3] Cooper, S. Dann, W., and Pausch, R. "Alice: a 3D Tool For Introductory Programming Concepts." Journal of Computing Sciences in Colleges, v 15, n 5. 2000 p107-116.

[4] Crews, T., and Ziegler, U. "The Flowchart Interpreter for Introductory Programming Courses." Proceedings of FIE '98, 307—312

[5] Gudmendsen, D. Olivieri, L., Sarawagi, N. "Using Visual Logic: Three different approaches in different courses – general education CS0, and CS1". Journal of Computing Sciences in Colleges v 26, n 6, 2011, p 23 – 29.

[6] Kulik, C. and Kulik, J.. "Effectiveness of Computer Based Instruction: An Updated Analysis". Computers in Human Behavior, v. 7 issue 1-2, 1991, pp. 75-94.

[7] Shackelford, R., and LeBlanc, R. "Introducing Computer Science Fundamentals Before Programming." Proceedings of FIE '97, 285—289

[8] Wenglinsky, H. “Does it compute? The relationship between educational technology and student achievement in mathematics.” Educational Testing Service Policy Information Center. 1998.

[9] Visual Logic. http://www.visuallogic.com. 2013.

Visualizing Problem Solving in a Strategy Game for Teaching Programming

Eshwar Bachu, Margaret Bernard
Department of Computing and Information Technology
University of the West Indies, St Augustine, Trinidad and Tobago

Abstract - *Lack of problem solving skill has been identified as the major cause of students' failure in introductory programming courses. This paper presents a strategy game, COPS, which helps students improve their problem solving ability by building program flowcharts as a jigsaw puzzle. For each move, pseudocode equivalents are immediately generated by the system. Students therefore have two visual representations of the problem solution and can more easily follow the logic in their solution. COPS provides dynamic feedback using visual and textual aids to guide students throughout the process. The results of two studies done with secondary school students in Trinidad and Tobago using COPS had positive results.*

Keywords: COPS, problem solving, visualization, educational game, flowchart, programming

1 Introduction

The high failure rates in introductory programming courses around the world are evidence of the difficulties which students face with computer programming. . Researchers have offered many reasons for students' failures such as poor mathematics skills and poorly designed instruction methodologies; however, low problem solving skill has been highlighted as their major challenge [1], [2], [3]. Students also have difficulty with program comprehension, program generation and programming fluency but their lack of problem solving skill remains their biggest challenge

Problem solving requires that students be able to understand the problem and what is required, determine the steps required to solve the problem and know the right order of the steps, and choose between different solutions. The main tools use during problem solving for the algorithm representation and development are flowchart and pseudocode. Pseudocode is a text based representation which consists of English-like statements. It is designed to fill the gap between the informal (spoken or written) description of the programming solution and the final program code [4]. Flowcharts are a visual representation of program flow using a combination of arrows and symbols to represent the actions and sequence of the program.

Games have been known to create interest, cooperation, and competition for its players. These are all qualities which educators have strived to inspire in the classroom [5], therefore it makes sense to merge the motivation of games with learning. Games have shown to alleviate some of the difficulties which students face while programming [6]; help increase their program comprehension and build programming fluency [7]; and avoid common syntax errors [8]. The success of games in teaching programming has also encouraged the use of game programming as a means of teaching programming [9].

During problem solving, the most challenging task for students is determining the right steps to solve the problem and knowing the right order of the steps. This paper presents a strategy game for the teaching and learning programming, which focuses on the visualizing of the problem solution. It uses flowchart and pseudocode with dynamic visual and text alterations to help students identify and follow the logic of the problem solution.

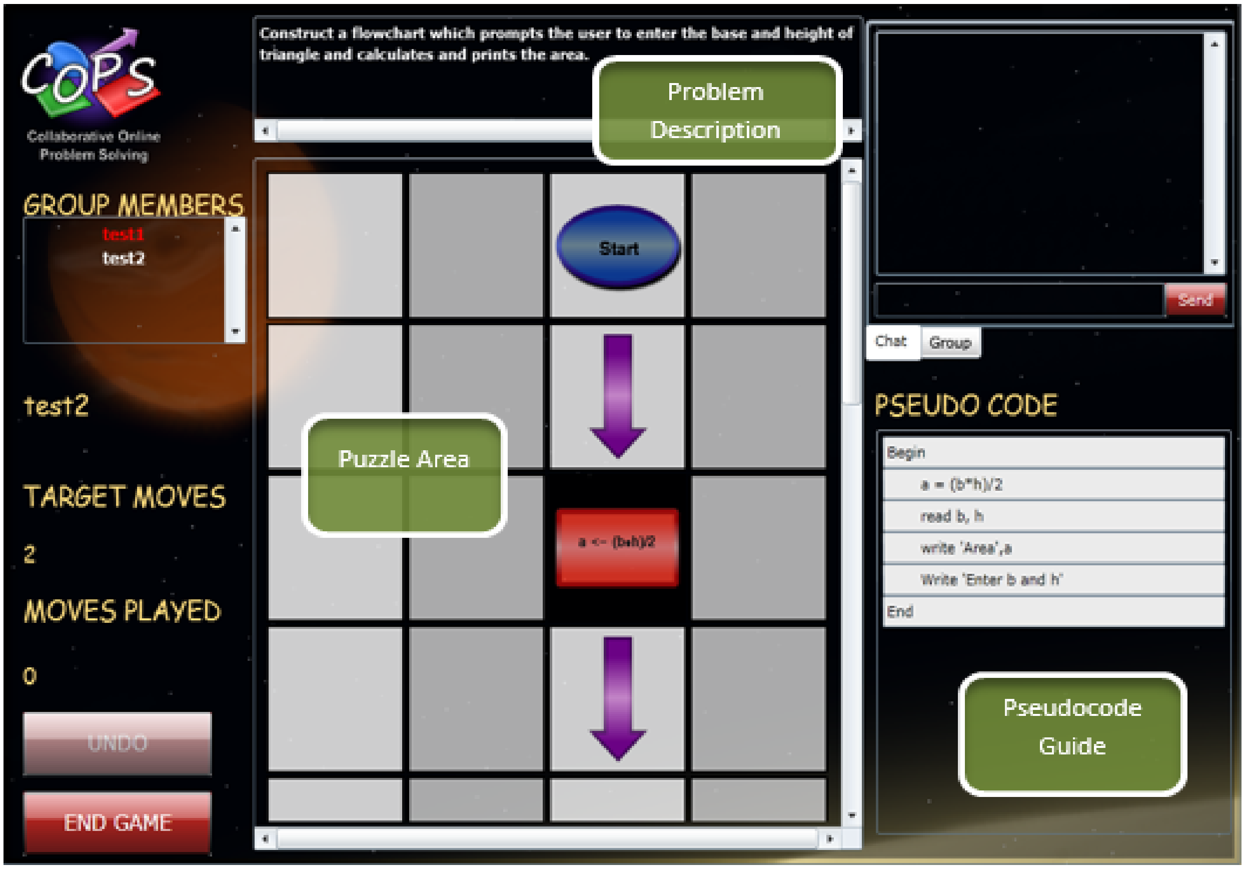
2 Collaborative Online Problem Solving (COPS)

COPS is a web based strategy game in which players are required to solve program flowchart puzzles. COPS has both single player and multiplayer modes but the game objective is the same. The multiplayer version was designed based on a framework presented by Bachu and Bernard [10]. The primary target users of COPS are adolescents between the ages of 13 and 17 since most students in Trinidad and Tobago and the wider Caribbean region are introduced to programming at the secondary school level. This suggests that students' difficulties need to be addressed at that level rather than at a university. Also, improving the situation at an earlier time would increase the likelihood of students being interested in pursuing higher education in computing.
Both flowchart and pseudocode are important tools for the teaching and learning of programming. Students have a greater preference for flowcharts [11] and flowcharts have been useful in the understanding of complex algorithms [12], while pseudocode helps students translate their solution in program code. COPS utilizes both flowchart and pseudocode.

2.1 Gameplay

There are two different puzzle types in COPS which are designed to challenge players differently:

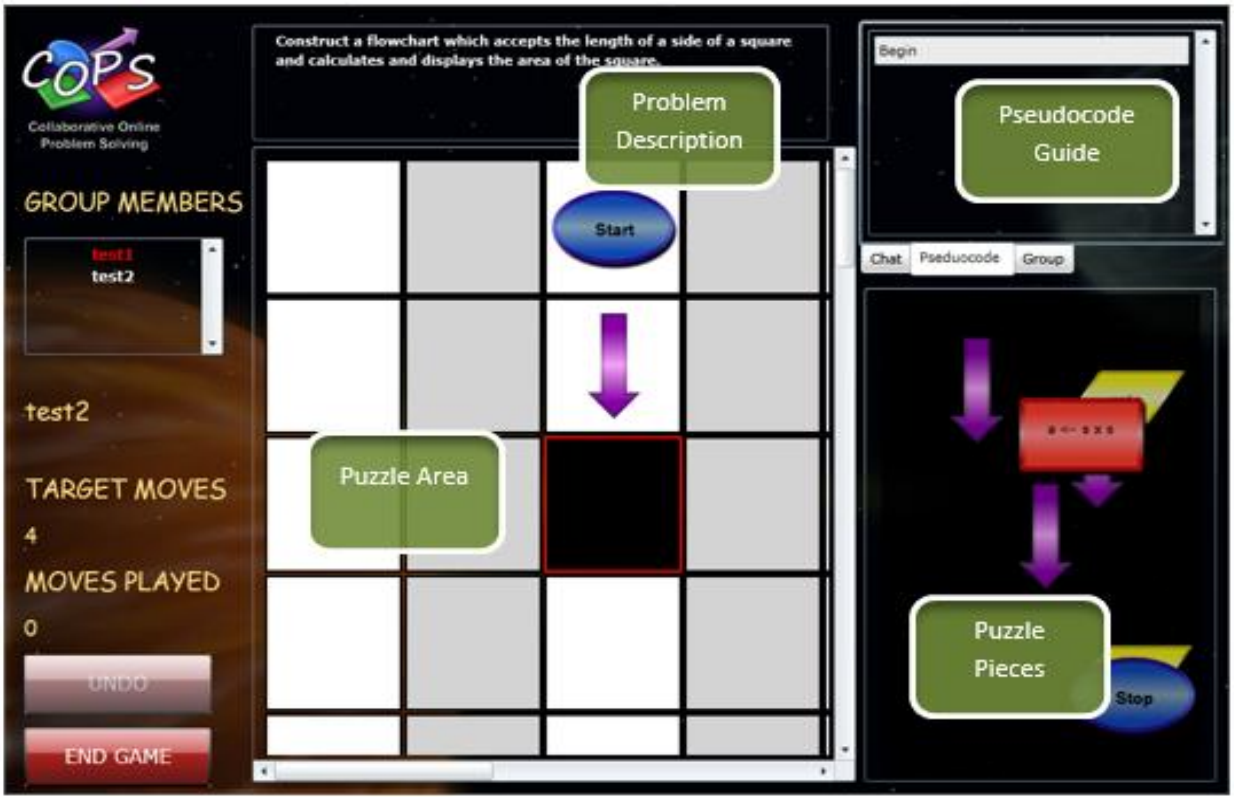
Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America



Construct a flowchart which prompts the user to enter the base and height of triangle and calculates and prints the area.

```
Begin
    a = (b*h)/2
    read b, h
    write 'Area',a
    Write 'Enter b and h'
End
```

Figure 1 COPS Multiplayer Swap Screenshot



Construct a flowchart which accepts the length of a side of a square and calculates and displays the area of the square.

Figure 2 COPS Multiplayer Jigsaw Screenshot

1. SWAP – The players are presented with a flowchart with pieces out of order and they are required to swap two pieces at a time to correct the flowchart. The players are given a target number of moves in which to solve the puzzle. This puzzle type was designed to be the easier of two to allow players to gain a sense of accomplishment so they are confident to attempt the harder puzzle type.
2. JIGSAW – The players are required to build a flowchart similar to building a jigsaw puzzle within a target number of moves.

Figures 1 and 2 show screenshots of COPS multiplayer swap and jigsaw games, respectively. The layout of the single player game is the same except that there are no group members, chat or group areas. In both modes, players are presented with a problem description and they are required to build the correct flowchart to solve the problem

Each puzzle in COPS has a difficulty level and a question type. The questions type corresponds to key concepts in introductory programming; arithmetic, if statements, if-else statements, while loops and while loops with if-else statements. Players are awarded points based on the number of moves in which the puzzle is solved. Players receive 10 points for each swap puzzle which they complete within the target number of moves and 20 points for jigsaw. One point is deducted for each move beyond the target. For games which they quit, no points are awarded. In the multiplayer version, group sizes are between two and four and one player is randomly chosen to make the first move and a turn base approach is used. All members of the groups receive equal points for a game. Players are allowed to play within different groups to increase their point accumulation.

2.2 Design of COPS

Games which are successful at teaching programming are those which cause the learner to develop and understand concepts from the content as a consequence of its system and interface [13]. All aspects of COPS have been designed to help students improve their knowledge and use of flowchart and pseudocode in problem solving. These tools have been used in classrooms for many years but most students have a preference for one or the other. COPS helps students develop a greater appreciation for how both can be used in designing the logic of the program solution as well as to understand how they complement each other. A space theme was used throughout COPS along with bright colors to make the game appealing to its target users.

The shapes used in the flowchart puzzles throughout COPS are the standard program flowcharts; a purple arrow for sequencing, a blue oval for start/stop, a yellow parallelogram for input/output, a red rectangle for process and a green diamond for decision. COPS has a question bank containing a large number of puzzles of varying types and difficulties. Each puzzle is chosen based on the current cumulative points of player and the difficulty level of the puzzles. All new players are allowed to attempt questions of difficulty level 1 (Very Easy); to attempt questions of increasing difficulty, their minimum point accumulation should be as follows: level 2 (Easy) – 50 points; level 3 (Intermediate) – 100; level 4(Hard) – 150 points; level 5 (Very Hard) – 250. The puzzles for the groups are chosen based on the average point accumulation for the members. At difficulty levels 1, 2 and 3, the jigsaw puzzles provide students with the exact number of pieces which are required to build the flowchart; however, at the harder levels, they are given additional pieces which do not form part of the solution. Game administrators or teachers can also choose to limit the puzzles to specific types. This feature is useful as a teaching tool where teachers may wish to limit the puzzles their students can attempt to the current topic they are teaching.

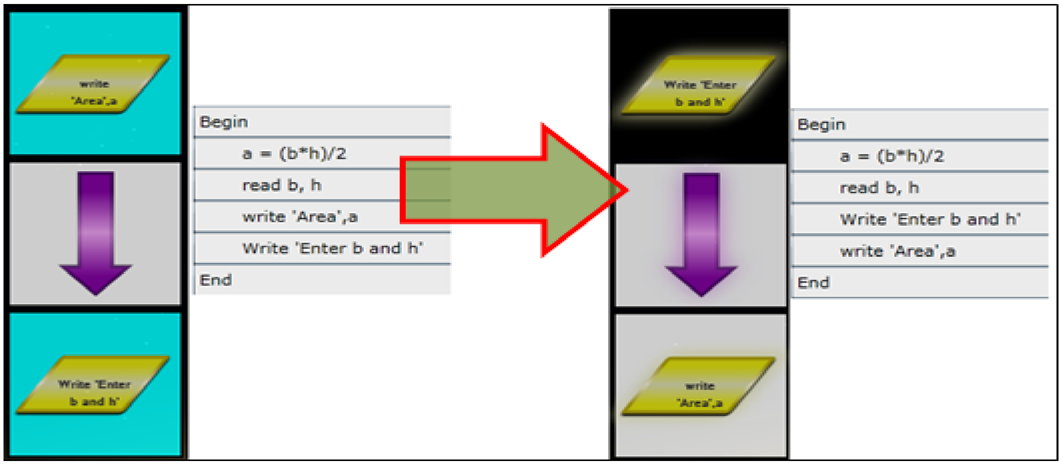
2.3 Visualizing Problem Solutions in COPS

The pseudocode in COPS is automatically generated according to the current flowchart regardless of whether the flowchart is correct. The pseudocode was presented for two purposes; to help players who get stuck with their puzzle and to show students how flowcharts can be translated into pseudocode. This automatic translation allows students to see how changes in program sequencing are represented in program code. This is a powerful visualization aid to their understanding the logic in their solution.

Beside the pseudocode, COPS provides intelligent visual feedback to players to assist them in solving the puzzle. While solving a swap puzzle, the pieces which are in incorrect positions have a black background in contrast to the general light grey/dark grey and when they have been corrected, the background changes to the normal colors. In the jigsaw puzzle, while the puzzle is being built, the correct portions of the puzzle which contain more than three pieces are greyed out to indicate to the players that it is correct to allow them to move onto other parts of the puzzle.

The possible positions where pieces can be placed are highlighted in a black background. The visual alterations which COPS provides are intended to guide the learning process and can also help encourage problem solving [14]. Students generally lose focus easily and get frustrated when they do not receive adequate guidance [15].

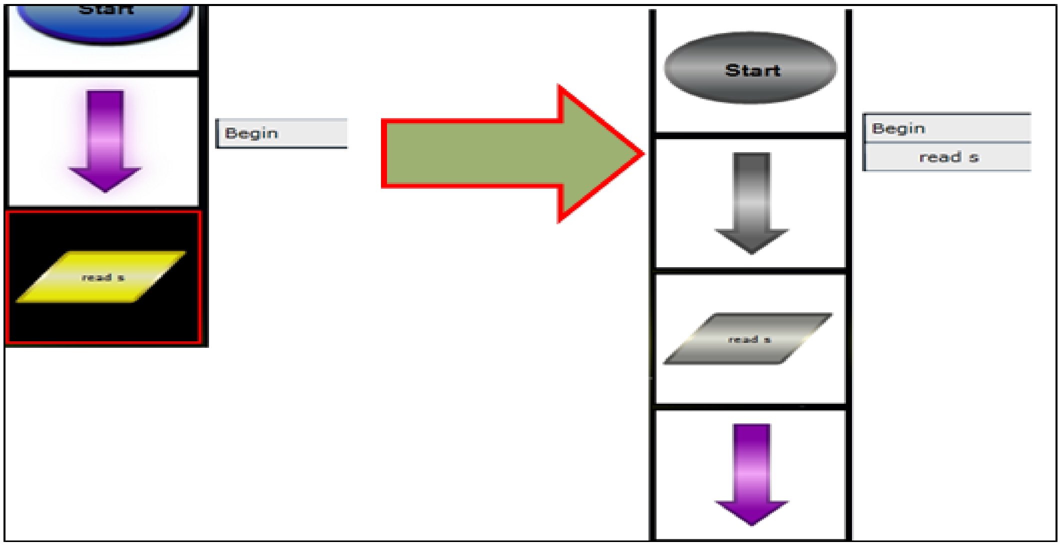
Figure 3 shows the pseudocode being updated when two pieces are swapped in swap puzzle game; the last two lines of the pseudocode are interchanged. Figure 4 shows the jigsaw update for a correct move and Figure 5 shows the jigsaw update for an incorrect move. In Figures 4 and 5, the pseudocode is still generated to match the flowchart but the non-greying of the newly



```
Begin
    a = (b*h)/2
    read b, h
    write 'Area',a
    Write 'Enter b and h'
End
```

```
Begin
    a = (b*h)/2
    read b, h
    Write 'Enter b and h'
    write 'Area',a
End
```

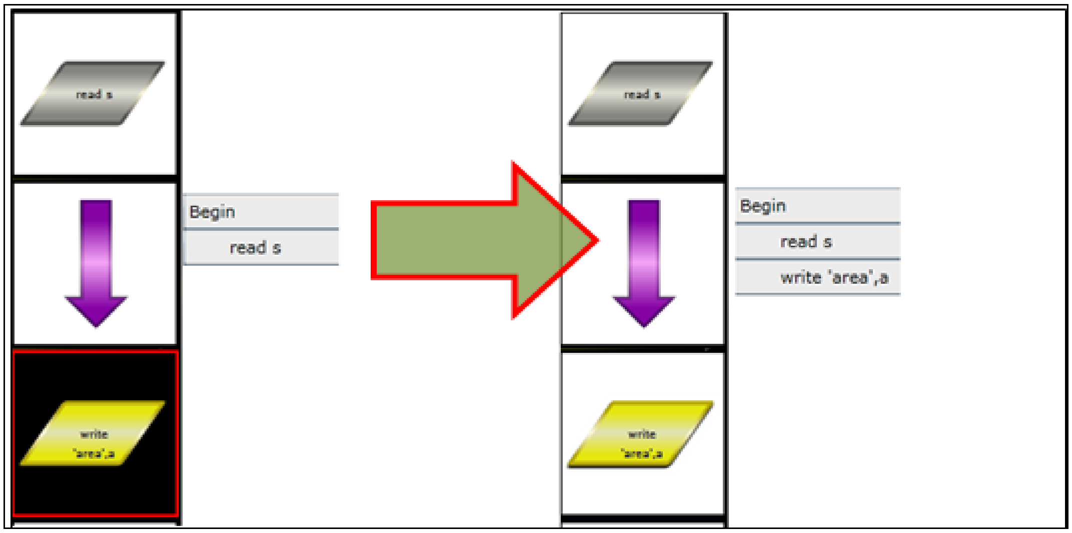
Figure 3 COPS Swap Puzzle Pseudocode Update



```
Begin
```

```
Begin
    read s
```

Figure 4 COPS Jigsaw Puzzle Correct Mode Update



```
Begin
    read s
```

```
Begin
    read s
    write 'area',a
```

Figure 5 COPS Jigsaw Puzzle Incorrect Move Update

added output symbol in Figure 5 indicates an incorrect move was made. During programming, the order of certain operations, such as variable initialization and input/output can be interchanged and COPS has been designed to accommodate this. When puzzles are being built into COPS, game administrators or teachers can specify which pieces or portions of the flowcharts can be interchanged and this allows players the flexibility to provide varying solutions to the same problem.

COPS stores data about every move made in all games, this allows the recreation of the games which can be used for analytical purposes by teachers and students. Additionally statistics such as the following can also be gathered from COPS:

- Performance on tasks by individuals and groups. This can be filtered by puzzle type or difficulty level

- Performance by individual players and groups. This can also be filtered by puzzle type or difficulty level.

With this information available, students can identify their strengths and weaknesses, reflect on their past experiences and improve their learning. Similarly teachers can use the information to structure their teaching.

2.4 COPS System Architecture and Implementation

COPS was built entirely using the .NET framework for easy integration and seamless functionality. The game client was developed between 2011 and 2012 and uses Silverlight 5.0. Silverlight was chosen due to its extensive user interface (UI) and animation possibilities. Silverlight is perfect for developing great looking and interactive web applications such as multiplayer games like COPS. The game server was developed with Windows Communication Foundation (WCF) Duplex Web Services. WCF is a framework for building service oriented applications and duplex communication was chosen to allow two-way communication between clients and server. C Sharp (C#) was the programming language used with Silverlight and WCF. NET.TCP is a TCP-based network protocol provided by WCF for Silverlight clients which facilitates high performance communication. The Database Management System (DBMS) used was SQL SERVER 2008 R2. LINQ TO SQL was used for all database processing. It is a component of the .NET framework which provides a runtime infrastructure for managing relational data as objects. All development was done using Microsoft Visual Studio 2010 Service Pack One, SQL Server Management Studio and Microsoft Expression Blend 4. The game's server was deployed on Internet Information Services Seven (IIS 7).

The COPS WCF web service was implemented as a singleton. This means that one WCF service instance was created for serving all clients where the different client requests are handled using multiple threads. The COPS Silverlight client was implemented using object-oriented design and a condensed class diagram is given in Figure 6. The visualization logic in COPS was implemented in the Game super class and its subclasses. Figure 7 shows a sequence diagram for each time a player makes a move. The details of each move are sent to the Game server and the details are stored in a database before it is sent to the other group members for multiplayer games. When each game client receives the move details, it updates the game accordingly. During the update, a dynamic programming algorithm is applied to identify the portions of the puzzle which are correct and the appropriate visualizations are applied.

3 Data Collection and Results

In August 2012 and January 2013, two studies were done to investigate the usefulness of COPS in helping to increase the problem solving abilities of novice programmers.
The first study had 27 participants from various secondary schools across Trinidad and Tobago, none of whom had done computer programming previously. All participants were required to attend 3 hour class sessions daily for two weeks. Since the students had no programming experience, the first three sessions were used to teach basic problem solving using traditional classroom methods. After the third session, an evaluation exam was administered (PRE-TEST); the participants were then introduced to the multiplayer mode of COPS and the remaining classes employed a blended learning approach where each session was spent using traditional teaching and COPS. The participants also had 24 hour access to COPS outside of the class sessions. Also, the puzzle types which the players were able to access were limited to the topic which was currently being taught. Another exam (POST-TEST) was administered at the end of the sessions. A paired t-test was done on the results of the pre-test and post-test which showed a mean improvement for of 14.27% (p-value=0.000).

The second study had 30 participants who had previously done computer programming at their various schools. A pre-test was administered and all participants were given 24 hour access to the multiplayer mode of COPS for three weeks, unlike the first study, there were no formal class sessions. The aim of the study was to investigate the usefulness of COPS as a revision tool for the topics they would have already covered at their respective schools. At the end of the three weeks, a post-test was conducted. The results of a paired t-test showed a significant mean improvement of 21.56% (p-value = 0.00).
Additionally, a two-sample t-test was done using results from study one and study two which showed that while the participants from the second study had a higher mean improvement, a p-value of 0.123 indicated that there was no significant difference between both groups. This finding suggest that COPS was just as beneficial as a learning tool to first time programmers as it was as a revision tool to those who previously done programming.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

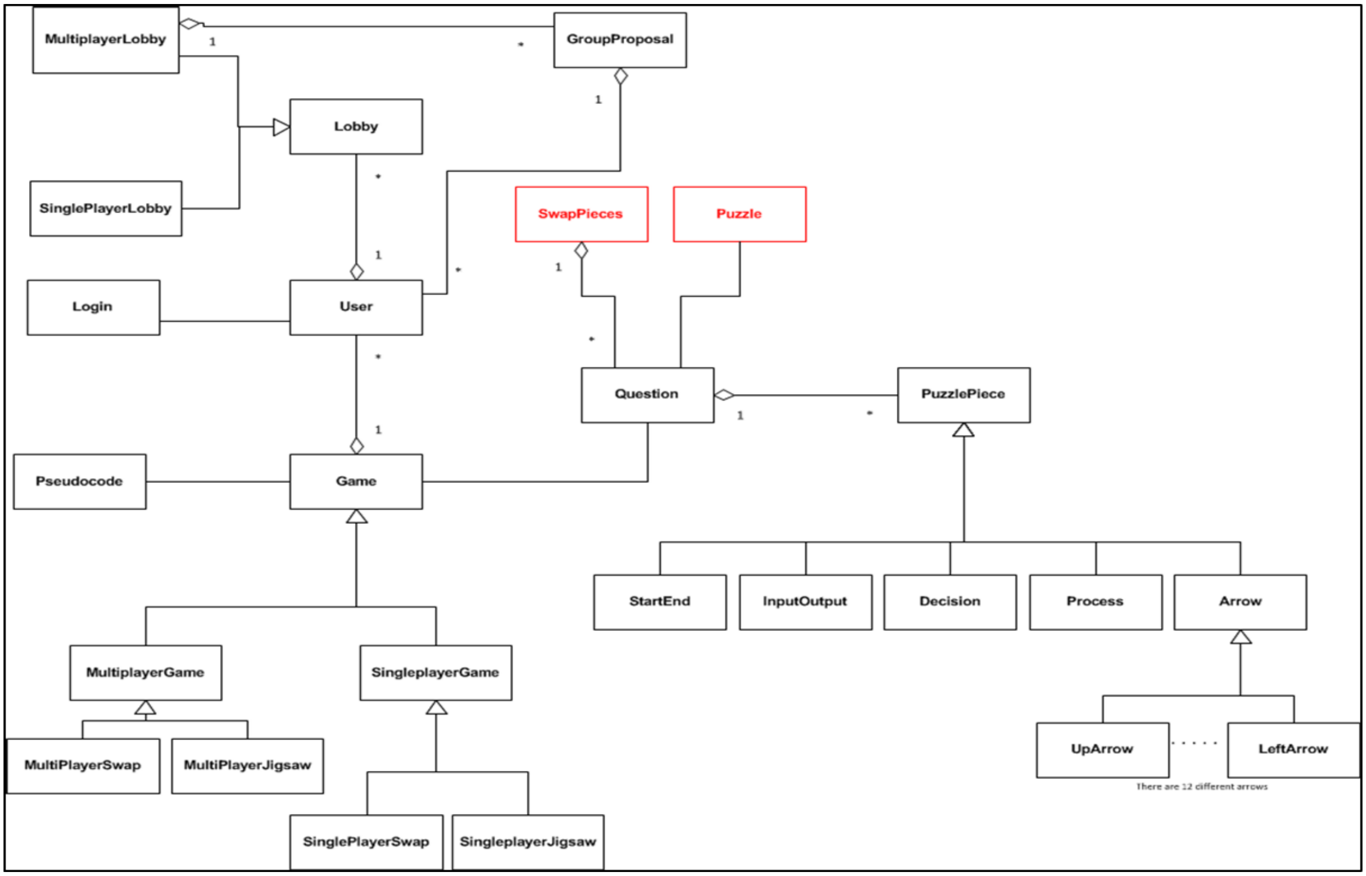


Figure 6 COPS Condensed Class Diagram

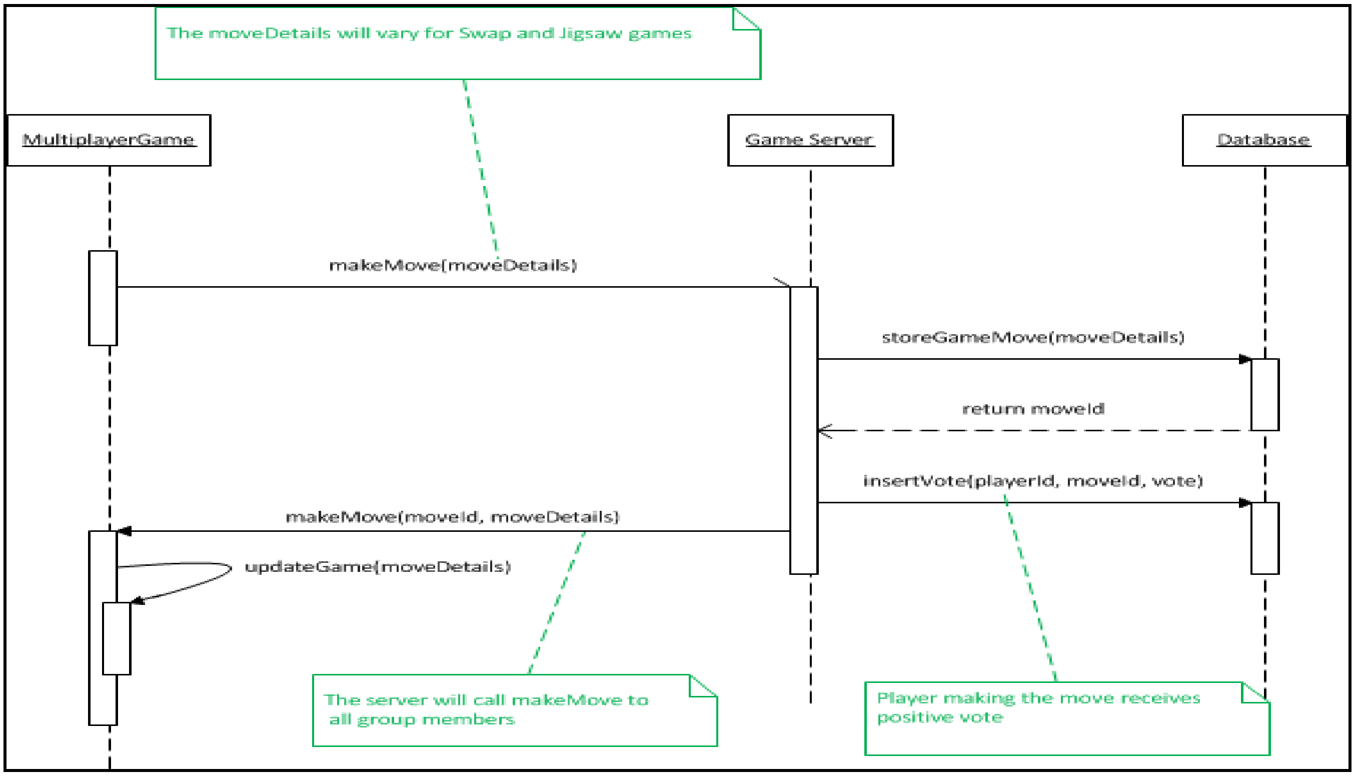

The moveDetails will vary for Swap and Jigsaw games
MultiplayerGame
Game Server
Database
makeMove(moveDetails)
storeGameMove(moveDetails)
return moveId
insertVote(playerId, moveId, vote)
makeMove(moveId, moveDetails)
updateGame(moveDetails)
The server will call makeMove to all group members
Player making the move receives positive vote

Figure 7 COPS Make Move Sequence Diagram

All participants from both studies were asked to complete a survey to provide feedback about COPS. Some of the major findings related to the visualization techniques were:

- 88% of participants found the automatic translation of the flowchart into pseudocode to be useful.
- 95% thought that the visual alterations in flowcharts helped them solve the puzzle faster.
- 97% thought that continued use of COPS would help them better understand flowcharts
- 89% thought that continued use of COPS would help them better understand pseudocode.

Data collected from COPS also allowed the identification of the type of puzzle which proved to be most difficult and to identify the specific program design problems which were the biggest challenge to students.

4 Conclusions

COPS is a strategy game designed to visualize problem solving using dynamic visual and text alterations. The visual problem solutions helped students to understand the logic in their solutions and the automatic mapping of the flowchart they had built, to its pseudocode equivalent reinforced that understanding. This learning took place in an engaging and motivating game environment. Qualitative and quantitative results from studies conducted have shown that COPS is beneficial to novice programmers. Currently COPS focuses on basic programming concepts but it can be easily extended to include more advanced topics.

5 References

[1] Theresa Beaubouef and John Mason. “Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations”. ACM SIGCSE Bulletin 37 (2): 103-106, 2005.

[2] S. Davies. “The Effects of Emphasizing Computational Thinking in an Introductory Programming Course”. In Proceedings of the 38th Annual Frontiers in Education Conference. IEEE. October 2008

[3] Anne L. Olsen. “Using Pseudocode to Teach Problem Solving”. Journal of Computing in Small Colleges 21 (2): 231-236. 2005

[4] Geoffrey G. Roy. “Designing and Explaining Programs with a Literate Pseudocode”. ACM Journal of Educational Resources in Computing 6 (1): 1-18. 2006

[5] Marc Prensky. “Digital Game-Based Learning”. Computers in Entertainment 1 (1): 1-4. 2003

[6] Rathika Rajaravivarma. “A Games-Based Approach for Teaching the Introductory Programming Course”. SIGCSE Bulletin 37 (4): 98-102. 2005

[7] Eshwar Bachu and Margaret Bernard. “Enhancing Computer Programming Fluency through Game Playing”. In Proceeding of the 4th Annual International Conference on Computer Games, Multimedia and Allied Technology (CGAT 2011). April 2011.

[8] E. M. N. S. Edirisinghe. “Teaching Students to Identify Common Programming Errors Using a Game”. Proceedings of the 9th ACM SIGITE Conference on Information Technology Education. ACM. October 2008.

[9] S. Leutenegger and J. Edgington. 2007. “A Games First Approach to Teaching Introductory Programming”. Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education. ACM. March 2007

[10] Eshwar Bachu and Margaret Bernard. “A Computer Supported Collaborative Learning (CSCL) Model for Educational Multiplayer Games”. In proceedings of the 11th International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, EEE’12, USA, July 2012

[11] Martin C. Carlisle, Terry A. Wilson, Jeffrey W. Humphries, and Steven M. Hadfield. “RAPTOR: A Visual Programming Environment for Teaching Algorithmic Problem Solving”. Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. ACM. February 2005

[12] D. A. Scanlan. “Structured Flowcharts Outperform Pseudocode: An Experimental Comparison”. IEEE Software 6 (5): 28-36.1989

[13] L. Doherty and V. Kumar. 2009. “Teaching Programming through Games”. Proceedings of the Technology for Education (T4E) '09. August 2009

[14] Jingyan Lu, Susanne P. Lajoie, and Jeffrey Wiseman. “Scaffolding Problem-Based Learning with CSCL Tools”. International Journal of Computer-Supported Collaborative Learning 5 (3): 283-298. 2010

[15] Sally He, Sally, Hong H. Kinshuk, and Ashok Patel. 2002. “Granular Approach to Adaptivity in Problem-Based Learning Environment”. Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT '02). IEEE. September, 2002

Guidelines for Implementing Pair Programming in Introductory CS Courses: Experience Report

Alex Radermacher, Gursimran S. Walia, Sameer Abufardeh, Oksana Myronovych

Department of Computer Science
North Dakota State University
Fargo, ND 58108
{alex.radermacher, gursimran.walia, Sameer.abufardeh, Oksana.myronovych}@ndsu.edu

ABSTRACT

Pair programming has been shown to be an effective method of improving the learning outcomes of students in introductory computer science courses. However, much of the existing literature related to pair programming does not focus how to effectively implement pair programming. Researchers studying multiple aspects of pair programming have conducted several empirical studies at our university over the past two years. During this time, researchers received valuable feedback from course instructors about the effects of implementing pair programming in their introductory computer science courses. These instructors also expressed concerns about the use of pair programming in their courses. These include being able to ensure equal participation from pair members and not being able to assess individual learning outcomes effectively. This paper reports these concerns and uses empirical evidence from the pair programming studies to provide guidelines for the effective use of pair programming in beginning programming courses. Based on the experiences at our university along with those experiences of other researchers, we provide recommendations for course design when using pair programming.

Keywords

Pair programming, CS1, CS2, course design.

1. INTRODUCTION

Pair programming began as part of Extreme Programming (XP), an agile development methodology proposed by Beck, et al. to help software developers be more productive [1]. Initial research showed that pair programming was effective when used by professionals [17]. Eventually researchers conducted studies that found the pair programming to be effective when used by novice programmers as well [13, 25].

Over the past two years, numerous studies related to pair programming has been conducted in introductory computer programming courses at our university. This research has produced valuable empirical evidence related to student learning, pairing strategies, and other aspects of pair programming (e.g., student-teacher interactions, students' mental model consistency etc). Throughout these studies, the primary researchers have worked closely with the three instructors who taught the classes in which the studies were conducted. These instructors have provided valuable feedback regarding the implementation of pair programming in introductory computer science courses to help researchers better understand the results, and on how to address the problems faced while designing a programming course when using pair programming.

Although pair programming has been demonstrated to be an effective method of improving student learning, there are several considerations that must be taken into account when using pair programming in an introductory computer science course. Examples include as how students will be paired and how to ensure that both pair members are contributing. Previous research has already provided many useful suggestions including strategies for effectively pairing students [5, 26], but there is a general lack of empirical evidence to guide course design when using pair programming.

This paper reports on instructor concerns regarding the use of pair programming and provides data from empirical studies conducted at several universities (including the studies conducted at the researchers' university) in order to provide a set of recommendations that can be used to most effectively implement pair programming in an introductory computer science course. Similar reports have been published by other authors in the past [3, 27, 28], but these papers mainly focus on problems experienced at a single university or were published several years ago and do not take recent advancements or discoveries into account.

The remainder of this paper is constructed as follows: Section 2 provides a brief description of pair programming and presents related work that forms the basis of this study. A description of the empirical studies on pair programming used to identify issues and solutions is given in Section 3. Section 4 lists instructor feedback and concerns which are validated by empirical evidence. Recommendations for remedying these problems are presented in section 5. Conclusions to this study are given in section 6. Finally, Section 7 briefly describes the ongoing and future research studies in pair programming at researcher's university.

2. BACKGROUND

Pair programming describes a programming technique where two programmers work together on the same programming task. One programmer, referred to as the driver, actively controls the keyboard and writes code, while the other programmer, referred to as the navigator, watches the driver for mistakes and helps him develop the code. The pair programmers work collaboratively while frequently exchanging the roles of the driver and the navigator.

The term pair programming was first used by Beck and his colleagues in *Extreme Programming Explained: Embrace Change* [1]; however, the effects of collaborative programming [17] as well as the use of collaborative learning in introductory computer science courses have previously been studied [21].

Substantial empirical studies were performed by Williams, et al. at NCSU (North Carolina State University) [24, 25] and McDowell, et al. at UCSC (University of California Santa Cruz) [3, 13, 14] that provided significant evidence to support the use of pair programming in introductory computer science courses, showing that it improved student performance on programming exercises and improved their enjoyment of programming. These initial findings provided good support for the use of pair programming in introductory computer science courses and motivated further research into pair programming.

Since then, additional empirical research has studied other aspects of pair programming such as the effects on student retention [7, 15], the effects of pair programming on female computer science students [2], and factors related to creating effective student pairs [5, 26].

Recent research conducted at NDSU (North Dakota State University) has provided further empirical support for the effectiveness of pair programming [20], studied newer alternative methods for pairing students that improved the pair performance [18], examined the effects that the use of pair programming has on student-instructor interactions during laboratory sessions [19], and are currently evaluating the effects of pair programming on mental-model consistency in introductory computer programming students.

The following section provides a more in-depth description and analysis of pair programming studies from which information has been extracted to support the research findings presented in later sections of the paper.

3. RELATED EMPIRICAL STUDIES

This section presents a more detailed description of the empirical studies on pair programming in an educational context and the major results from those studies. This is done to highlight that the identified issues with pair programming implementation (as discussed in Section 4) and the proposed solutions to those issues (as discussed in Section 5) are based on a large collection of empirical evidence across different research sites.

Inclusion and Exclusion Criteria: Papers included in the analysis were those that reported the results from an empirical investigation of pair programming or using the results of previous empirical studies to form conclusions. Some studies were excluded from the analysis as they included participating subjects who worked together remotely or because the participating subjects were not students. Also, those studies whose findings were unclear or ambiguous were excluded from the analysis.

Many papers present results or analysis based on the same data sets or share common authors. Because of this, papers are grouped according to the university site where the research was conducted.

3.1 NCSU Studies

A large number of studies related to pair programming have been conducted at NCSU over the course of the last decade [12, 16, 22, 25, 26, 27, 28].

The initial studies conducted at NCSU were part of an NSF (National Science Foundation) supported longitudinal study [25]. During the Fall 2001 semester, two sections of the introductory programming course were included in the experiment. The study required 69 subjects in one section to work individually and served as a control group, whereas 44 subjects in the other section served as the experimental group and used pair programming. The focus of this study was to gain a better understanding of how pair programming affected student learning and performance. Quantitative results from the study indicated that pair programming improved student performance on programming exercises, but that pair programming did not improve student performance on exams. The study also included qualitative findings related to student behavior when using pair programming and role of instructors during laboratory sessions. One of the most notable reported observations was that when pair programming was used in programming labs, students appeared to spend less time waiting for assistance from an instructor.

A follow-up study that added additional data points was conducted during Spring 2002 semester [16]. An additional 102 subjects worked individually as part of a control group and 280 subjects used pair programming. As with the previous study, only the results of freshman and sophomore students who took the course for credit were analyzed. The results of this study indicated that pair programming was effective for improving the learning outcome for non-computer science majors, but did not have a significant impact on computer science majors. The study also provided further support that the use of pair programming did not improve student performance on exams. The study also provided qualitative analysis that stipulated that random pairing led to incompatible pairs in a small percentage of cases and that lab instructors should monitor pairs to ensure that pair programming is being used properly.

Katira, et al. conducted a study during the Fall 2002 and Spring 2003 semesters to study factors influencing pair compatibility [11, 12]. The study analyzed 564 students from three courses for pair compatibility based on personality, skill level, perceived competence, and self-esteem. There was moderate support to indicate that partners with different personality types were more compatible and that pairs were more effective if both partners believed that they had similar levels of technical competence. However, the researchers did not find that the pair compatibility was dependent on actual skill level (measured by performance on a midterm exam).

Shrikanth, et al. reported on student and instructor perceptions of the viability of pair rotation [22]. Subjects in this study were 240 students in a CS1 course and 170 students in an undergraduate SE (software engineering) course, which were a subset of the subjects from the experiment conducted by Katira [11]. All participating subjects used pair programming and were assigned four partners over the course of the semester. The researchers concluded that pair rotation had benefits for both students and instructors, but also pointed out several issues that could arise when constantly adjusting student pairs. One prominent problem identified by the authors is that mandatory pair rotation could break apart a functional pair and result in new pairs that were less effective. The researchers noted that because only a small number (17) of subjects from the SE course completed the survey, the results may be slightly biased.

Williams, et al. released a follow-up study to further examine compatibility of student pair programmers [26]. This study used data collected by Katira [11] but included 133 additional subjects in a software engineering course in the Fall 2004 and Spring 2005 semesters. Researchers examined the impact that learning styles, work ethic, and time management skills had on pair compatibility. Results of the study indicated that pairing Myers Briggs sensors with Myers Briggs intuitors, classifications (used to describe how people perceive and understand information) produced more compatible pairs and that pairing students with similar work ethics also produced more compatible pairs.

Williams also released two papers that provided a list of lessons learned from using pair programming [27] and suggested guidelines for using pair programming [28]. These papers were based on the results of previous studies conducted at NCSU. A few of the major points not discussed in previous work include providing students with training in order to help them understand how to effectively use pair programming. Both papers also expressed the importance of ensuring that instructors or other teaching staff are engaged in managing pair interactions.

3.2 UCSC Studies

Several early studies of pair programming were conducted at USCS by McDowell and colleagues [3, 13, 14, 15].

The initial studies conducted at UCSC evaluated the effectiveness of pair programming on the performance and retention of female students in computer science or related fields [13, 15]. Subjects for the study were students enrolled in two different sections of the introductory programming course during the 2000 – 2001 academic year. In this study, 172 subjects from one section completed programming assignments in pairs, while 141 subjects from the other section work independently. The results of this research indicated that pair programming improved student grades on programming exercises, but did not improve student grades on exams. The authors also noted that the use of pair programming may lead to fewer students dropping the course, but did not have a sufficient number of data points to determine if this result also applied specifically to female students.

McDowell, et al. reported on another study focused on examining the effects of pair programming on student persistence, perception, and performance [14]. Subjects in this study were 555 students enrolled in the introductory programming class in the 2000 – 2001 academic year, and also included the subjects from the previous study. The major findings of this study were that students who use pair programming indicated that they were more confident in their work, were more satisfied with programming, and had greater enjoyment than students who worked independently.

Beven, et al. released an experience report, providing guidelines and suggestions for successfully implementing pair programming [3]. The recommendations were based on lessons learned and issues experienced with implementing pair programming during the 2000 – 2001 academic year. The researchers reported the main sources of difficulty were due to student scheduling conflicts and large disparities in skill level. The authors of the study also presented suggestions for course design (e.g. designing lab exercises that can be completed with minimal out-of-class time requirements.) and pair arrangement, indicating that students of somewhat similar ability should be paired together.

3.3 NDSU Studies [Researchers' Site]

Recently, a series of studies related to pair programming were conducted at NDSU [18, 19, 20].

Radermacher et al., reports on the results of two different studies conducted during the Spring 2010 semester [18]. Subjects in the first study were 35 students enrolled in one section of the CS1 course and the second study included 39 students enrolled in two sections of the CS2 course. Subjects in the CS1 course were split into two groups, one which used pair programming and one that did not; whereas the subjects in the CS2 course were paired based on declared major. Researchers reported that subjects from both the CS1 and CS2 courses indicated that they felt pair programming improved their understanding of programming concepts. Another major result indicated that pairing a computer science (CS) student with a non-computer science (nonCS) student produced less compatible pairs as compared to CS-CS pairs and nonCS-nonCS pairs.

Radermacher et al., reported another empirical study that investigated the effects of pair programming on student-instructor interactions during programming laboratory sessions [19]. Subjects in this study were 44 students enrolled in one section of the CS1 course and 53 students enrolled two sections of the CS2 course during the Fall 2010 semester. Subjects in the CS1 course alternated between using pair programming and working individually during lab sessions, whereas subjects in the CS2 course only used pair programming. Researchers monitored these lab sessions, marking the number of questions asked, how long it took before an instructor could address the subject's question, and how long the instructor spent interacting with the subject. Results of the study indicate that when pair programming is used, students spend less time waiting for assistance from an instructor and spend more time interacting with the instructor, likely due to a decrease in questions related to syntax errors or other minor problems.

An ongoing experiment at NDSU investigated the effects of pairing subjects based on their mental model consistency levels (ranging from highly consistent to highly inconsistent) at the beginning of the semester to evaluate changes in the students' mental model consistency and their programming performance [20]. The initial evidence suggest that such a pairing strategy can be an effective way if previous performance data is not available and that certain mental-model-based pairing arrangements (and not all) are more effective in migrating students towards greater consistency and resulted in better performance on exams.

3.4 Dickinson College Studies

Braught, et al. reported a study examining the effects of pairing students based on ability [5]. Subjects in this study were 259 students enrolled in 13 different sections of an introductory programming course taught between 2005 and 2008. 142 subjects were pair based on ability, 41 were paired randomly, and 72 were not paired. Subjects in this study completed weekly programming assignments and completed 5 exams, two of which were programming exams where students produced code to complete a program. The results from this study indicate that students in the lower quartile who were paired by ability performed better on programming exams than students in the lower quartile who were paired randomly or not paired at all. This contrasts earlier research findings from studies conducted at NCSU and UCSC, which did not find any significant improvement for students on exams when using pair programming.

Braught, et al. also examined how the results of their studies compared with the results from several previous studies conducted by other researchers [6]. The results of this report provided additional support for the conclusion that pair programming provided benefits for students at lower SAT levels. The researchers also provided qualitative data supporting the conclusion that using pair programming in programming labs resulted in fewer questions related to syntax issues or other lower-level problems.

3.5 University of Sussex Studies

Chaparro, et al. examined factors that affected student perceptions of the effectiveness of pair programming [9]. Subjects in the study were 80 post-graduate students enrolled in an object oriented programming course during the Fall 2004 semester. Researchers found that subjects preferred to work with someone who was similarly skilled or more skilled than they were and that subjects felt that pair programming was not useful if the programming task was not challenging. Observational evidence also suggested that when a large skill disparity existed between subjects, the more skilled subject would take control and relegate the less skilled subject to a more passive role.

4. PROBLEMS AND CONCERNS WITH THE USE OF PAIR PROGRAMMING

This section details problems encountered by instructors when using pair programming in an introductory computer science course. Both issues which have been encountered at NDSU and issues which have been described in previous research papers are considered. Although these problems and concerns are broken into five different categories, there is often a relationship between them, such that issues experienced in one area often exacerbate problems that are experienced in another area.

4.1 Individual Assessment

Difficulties assessing individual learning and ensuring that both partners are benefiting from the use of pair programming is a common issue with pair programming. One of the CS1 instructors at our university also expressed concerns that several students, who he felt would not have normally been able to pass the class, had been able to pass the class because their partner was able to help carry them. Table 1 shows the percentage of students passing the course when pair programming was used and when students worked individually based on eight semesters of historic data for this instructor's class.

This shows a large increase in the number of students who were able to pass the course. The results are even more skewed than they appear as the drop-rate for the course during the Fall 2010 semester was slightly more than 8% compared to the historical drop-rate of approximately 18%. Bevan, et al. had also reported that students were willing to submit assignments that only one of the students had completed [3]. Williams, et al. also expressed similar concerns when they discovered instances of students who performed well on pair programming exercises, but scored poorly on exams, suggesting that one partner may have been completing most or all of the work on the programming exercises, a phenomenon that was also observed by our own instructors. Another of our instructors indicated that this may have been an issue as a large number of laboratory assignments required substantial out-of-class work in order to complete and that there was no easy way to ensure that both partners had contributed equally.

4.2 Subject Pairing

Another issue regarding pair programming is ensuring that individuals are paired effectively, especially early in the semester. Previous research has indicated that it is not feasible to match students based on existing available measures such as SAT score, GRE, or GPA [26]. Nagappan, et al. reported that random pairing lead to conflict and undesirable pairs so it is preferable to avoid pairing students in this manner [16]. It was also reported that pairing computer science students with non-computer science students produced less compatible pairs [18].

Table 1. Pre and Post Pair Programming Result Comparison of Student Grades

Percentage of Students Receiving Grade or Better	*Historical Average*	*Fall 2010*
Percentage of students with 'C' or better grade.	68.87%	74.68%
Percentage of students with 'D' or better grade .	80.21%	93.57%

Another important aspect related to effective pairing is ensuring that student pairs will be able to work together outside of the class room. One of our instructors mentioned that some of the laboratory assignments were too large to be completed during laboratory periods. In these cases, it is necessary to ensure that students will be able to meet outside of class to work on the assignment together. Bevan, et al. also described this issue, but indicated that students may not always report scheduling issues or conflicts with their partner [3].

Another important question to ask is how frequently pairs should be rearranged, if at all. Because some pairs will experience scheduling conflicts or other problems that make the pair ineffective, some amount of rearrangement will be necessary. Shrikanth, et al. indicated that while there were advantages of frequently exchanging pairs, a large number of students indicated that it took time to become adjusted to working with a new partner and that in some cases, mandatory rearrangement destroyed an existing, highly compatible pair [19].

4.3 Assignments and Grading

One of the major issues with pair programming that we experienced was related to assignments and grading. One of the CS1 instructors found that when pair programming was used in his course the student grades became much less diverse, making it more difficult to assign letter grades based on performance.

Figure 1 shows the distribution of student grades when pair programming was used and compares it against the student grades from eight semesters of historic data. A chi square test indicated that the distributions were significantly different ($p = .014$). Additionally, drop rates for the course were significantly lower ($p < 0.001$), and the student grades were spread over a much narrower range when pair programming was used.

This is likely related to issues discussed in 4.1 where it was reported that one student may be carrying the other. McDowell, et al. also discussed the possibility of grade inflation, i.e. that higher grades are a result of one member carrying the team [13]. Based on analysis of student performance at our university, it appears as though this was the cause of the change in grade distribution.

Chaparro, et al. also discussed the necessity of ensuring that programming assignments are adequately challenging [9]. Students in that study indicated that pair programming was not as beneficial if the assigned programming task was trivial or quickly completed. One of our instructors reported a similar issue for the last assignment of the semester, which was designed to be considerably easier than previous assignments. He noted few

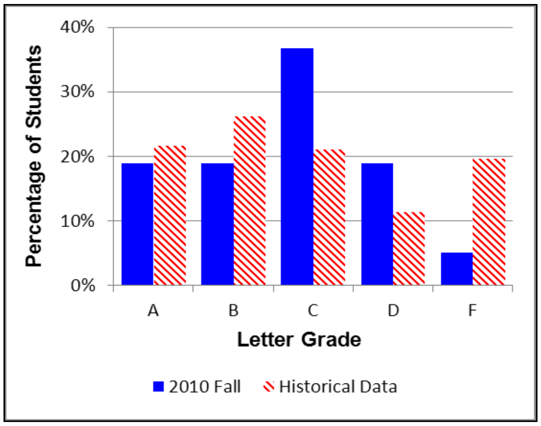


Figure 1. Comparison of Distribution of Student Grades Prior to and After Implementing Pair Programming

cases where one member of the pair would individually complete the exercise within the first day of it being assigned without consulting his or her partner.

4.4 Pair Interaction

Multiple researchers have indicated issues with how students interact. Williams, et al. found that students often did not properly use pair programming [25]. The most common identified issues from that research were that some pairs did not exchange roles at all while pair programming. Researchers at NCSU also indicated that students were unlikely to change roles unless instructed to during lab sessions. Chaparro, et al. did note that students should be allowed some leeway in role-switching and that forcing students to exchange roles at set intervals could interrupt students [9].

Our experiences were similar. In instances where the instructor took an actively role in telling pairs to switch, students were more likely to exchange roles, but if they were not instructed to do so, it was not uncommon for students to maintain whichever role that they had assumed for the duration of the session.

4.5 How Much Pair Programming

All three of the instructors with whom we worked have expressed some concern that using pair programming for every assignment may not be as effective as only using it for some assignments. One of the CS1 instructors felt that most new students did not have any programming experience and that until they gained some knowledge they wouldn't be able to effectively pair. One instructor noted instances where one member of a pair would actually be providing the other with incorrect information.

Another instructor felt that some assignments should be completed individually in order to provide the opportunity for students to show that they have mastered the ability to program without the need of a partner. The instructor stated that occasionally one partner would be absent from a lab and that the remaining student seemed to struggle, even though that student had done well on previous lab exercises.

Previous research has also indicated that some students feel that it is important to work individually [20].

5. SOLUTIONS AND RECOMENDATIONS

Based on our own experiences along with recommendations and results from previous research, we present a list of possible solutions for the problems and issues outlined in the previous section. It is believed that these suggestions are useful and valid as they are based on over a decade of empirical studies conducted on pair programming.

5.1 Individual Assessment

Our instructors felt that there were multiple ways to approach this issue. One instructor felt that having students complete lab assignments using a mixture of pair programming and individual work would provide an adequate amount of information to determine that students are able to program. When using a laboratory format where students alternated between using pair programming and working individually for each programming exercise, this instructor felt that there were not students who were able to take advantage of a strong partner in order to pass the class.

Another instructor indicated that having short in-class quizzes of programming concepts related to the previous lab would help to determine if each member of a pair was learning and benefitting equally. There were some concerns that this would create an excessive amount of work, but it was suggested that using a student response system could reduce the necessary amount of work on the part of the instructor and other researchers have reported that using these systems have other positive impacts in introductory programming courses [8].

In their initial studies, Williams, et al. implemented a policy where only students who scored above a certain threshold on exams were allowed to work in pairs [25]. This was done in order to prevent cases where an unskilled individual could get by with having a strong partner. This solution may not be ideal, especially given the recent findings that pair programming is most beneficial to students in the lowest quartile and improves their performance on programming exams [5]. This evidence contradicted most existing evidence regarding student performance on exams, although the authors were careful to distinguish that this was only observed on programming exams where students wrote code, and not the more traditional, written exams.

5.2 Subject Pairing

Subject pairing can be somewhat difficult. Williams, et al. indicated that some effective methods do exist [26, 27]. The most effective methods were pairing based on midterm score, pairing a Myers Briggs sensor with a Myers Briggs intuitor, or pairing students who share a similar work ethic.

Pairing based on midterm grade is impossible at the beginning of a CS1 course, but previous exam scores could be used to pair students taking a CS2 course. There is a general consensus among researchers that pairing students who are similar in ability, perceived or actual, is effective [5, 12]. Radermacher, et al. are investigating using mental model consistency tests to assess students and create pairs based on similarities in mental model consistency [20]. These tests which were developed by Bornat and Dehnadi can be given to students without previous programming experience and provide a reasonable assessment of ability [4, 10]. Other possible tests described by Simon, et al. could also be used to create ability-based pairs at the onset of the semester [23]. However, the effectiveness of such pairing methods have not been well examined.

Considering that all of our instructors also felt that it was important for students to complete some assignments individually, another possibility is to have students complete several initial programming exercises individually before assigning them to pairs. This approach provides data which can be used to construct ability-based pairs.

5.3 Assignments and Grading

Previous studies have provided several different suggestions regarding assignments. Chaparro, et al. indicated that it is important to create programming exercises that were sufficiently challenging, otherwise students were less likely to view pair programming as beneficial [9].

Guidelines published by Beven, et al. indicated the importance of producing programming exercises as a function of class time [3]. The authors stipulate that programming exercises should be designed so that the vast majority of students will be able to complete the assignment during scheduled laboratory sessions.

Williams, et al. suggest balancing the grades so that individual work has a larger impact on the overall grade than pair work [28]. In the introductory programming class at NCSU, pair work only accounts for 10% of the total grade. Another policy for other courses where pair programming was used is that students must have a passing grade on the individual portions of the class in order to receive a passing grade [27]. This solution allows pair programming to benefit a student, but does not allow them to pass unless they've shown sufficient individual mastery of the course content.

5.4 Pair Interaction

Multiple researchers have indicated that a closed, mandatory laboratory session is necessary for pair programming to be effective [3, 27]. This has multiple benefits because it allows instructors to monitor students and alleviate the lack of role switching [25]. It can also reduce the frequency of scheduling conflicts [3]. Additionally, this affords instructors an opportunity to reinforce good pair programming behavior by reminding students to frequently switch roles.

In courses where a substantial amount of work must be completed outside of the course, Williams, et al. indicate the importance of using a peer review system [28]. This allows instructors to identify dysfunctional pairs where partners are either unable to find sufficient time to work together or where one partner is completing a majority of the work.

5.5 How Much Pair Programming

There is no doubt that pair programming is beneficial to student learning [5, 25] and enjoyment [14] and has benefits for instructors as well [19]. It has already been suggested that pair work should not constitute the majority of a student's grade, but has not been suggested how much pair activity should be conducted.

In introductory computer science courses where a majority of subjects have little or no experience with programming, our instructors feel that they should spend some initial time at the beginning of the course working individually in order to acquire some programming knowledge before working with a partner.

Such an arrangement has several benefits. First, it provides performance data that can be used to pair based on ability, as most researchers agree that this is the most desirable pairing strategy [3, 5, 26]. Secondly, it allows a period of time for students who do not intend to complete the course to drop before pairs are created, minimizing the need to re-pair students. Additionally, initial programming assignments are more likely be trivial or straightforward and may not benefit from the use of pair programming [9].

6. CONCLUSION

Pair programming is an area where there are still many unanswered questions, but there is a consensus among researchers that the use of pair programming is beneficial for multiple reasons. However, there are many important considerations to keep in mind when implementing pair programming for the first time.

Based on the experiences at our university along with study results and recommendations from previous researchers, it is suggested that when using pair programming for the first time, the following guidelines should be followed in order to ensure the experience is beneficial and enjoyable for students and instructors:

- Subjects should be paired such that they have similar levels of ability and availability. This can be accomplished by requiring students to complete some minimal amount of work individually to determine programming ability. The use of mental model consistency or other tests may also be used to establish a reasonable baseline [4, 23].
- Mandatory laboratory sessions are essential in order to ensure that students are appropriately using pair programming and that there is at least some time that both partners can easily meet.
- Programming exercises should be designed such that they do not require substantial amounts of out-of-class time in order to complete. It should be largely possible for students to complete programming exercise assignments during scheduled laboratory sessions. This is suggested in order to maximize the amount of time spent in a location where appropriate pair programming behavior can be reinforced by an instructor and to minimize the amount of time where students can engage in undesired behavior.
- Pair work should not constitute a majority of a student's grade, but students should still be graded on their ability to write code. Using programming exams allows instructors to grade more heavily on programming ability and also provides an opportunity for additional individual assessment.
- When using pair programming in introductory courses, pairs should be re-arranged. This exposes students to a larger number of classmates and helps them make acquaintances, whom they will likely work with in future classes. It also helps alleviate problems with ineffective pairs and ensures that no one student will be stuck with a bad partner.

These guidelines, while useful, should not be considered concrete. In some cases, there is not sufficient empirical data to provide strong support for one particular point of view. Additionally, every university faces different conditions and constraints.

7. ONGOING AND FUTURE WORK

There is no doubt that pair programming is beneficial to student learning [5, 25] and enjoyment [14] and has benefits for instructors as well [19]. One question which does not contain a significant amount of empirical support is to what extent pair programming should be used in introductory computer science courses. In the future we plan to investigate this by studying student learning outcomes when pair programming is only used partially. Two possible approaches are to only use PP during the second half of the semester, and to alternate between using pair programming and working individually

8. REFERENCES

[1] Beck, K. 2000. Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading, MA, USA.

[2] Berenson, S.B., Slaten, K. M., Williams, L., and Ho, C. 2004. Voices of women in a software engineering course: reflections on collaboration. *Journal on Educational Resources in Computing.* 4, 1, (March 2004), Article 3.

[3] Bevan, J., Werner, L., and McDowell, C. 2002. Guidelines for the Use of Pair Programming in a Freshman Programming Class. In *Proceedings of the 15th Conference on Software Engineering Education and Training* (CSEET '02). IEEE Computer Society, Washington, DC, USA, 100-108.

[4] Bornat, R., Dehnadi, S., and Simon. 2008. Mental models, consistency and programming aptitude. In *Proceedings of the tenth conference on Australasian computing education* (ACE '08), S. Hamilton and M. Hamilton (Eds.), Vol. 78. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 53-61.

[5] Braught, G., MacCormick, J., Wahls. T. 2010. The benefits of pairing by ability. In *Proceedings of the 41st ACM technical symposium on computer science education* (SIGCSE '10). ACM, New York, NY, USA, 249-253.

[6] Braught, G., Wahls, T., Eby, L. M. 2011. The Case for Pair Programming in the Computer Science Classroom. *ACM Transactions on Computing Education*, 11, 1 (February 2011), Article 2.

[7] Carver, J. C., Henderson, L., He, L., Hodges, J., and Reese, D. 2007. Increased Retention of Early Computer Science and Software Engineering Students Using Pair Programming. In *Proceedings of the 20th Conference on Software Engineering Education & Training* (CSEET '07). IEEE Computer Society, Washington, DC, USA, 115-122.

[8] Chamillard, A. T.. 2011. Using a student response system in CS1 and CS2. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (SIGCSE '11). ACM, New York, NY, USA, 299-304.

[9] Chaparro,, E.A, Yuksel, A., Romero, P., and Bryant, S. 2005. Factors affecting the perceived effectiveness of pair programming in higher education. In *Psychology of Programming Interest Group 17th Workshop*, 5-18.

[10] Dehnadi., S. 2006. Testing programming aptitude. In *Proceedings of the Psychology of Programming Interest Group 18th Annual Workshop,* 22-37.

[11] Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., and Gehringer, E. 2004. On understanding compatibility of student pair programmers. In *Proceedings of the 35th SIGCSE technical symposium on computer science education* (SIGCSE '04). ACM, New York, NY, USA, 7-11.

[12] Katira, N., Williams, L., and Osborne, J. 2005. Towards increasing the compatibility of student pair programmers. In *Proceedings of the 27th international conference on Software engineering* (ICSE '05). ACM, New York, NY, USA, 625-626.

[13] McDowell, C. Werner, L., Bullock, H., and Fernald, J. 2002. The effects of pair-programming on performance in an introductory programming course. *SIGCSE Bulletin.* 34, 1 (February 2002), 38-42.

[14] McDowell, C., Werner, L., Bullock, H., and Fernald, J. 2003. The impact of pair programming on student performance, perception, and persistence. In *Proceedings of the 25th international conference on software engineering* (ICSE '03), IEEE Computer Society, Washington, DC, USA, 602-607.

[15] McDowell, C., Werner, L., Bullock, H., and Fernald, J. 2006. Pair programming improves student retention, confidence, and programming quality. *Communications of the ACM*, 49, 8 (August 2006), 90-95.

[16] Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., and Balik, S. 2003. Improving the CS1 experience with pair programming. *SIGCSE Bulletin.* 35, 1 (January 2003), 359-362.

[17] Nosek, J. The case for collaborative programming. 1998. *Communications of the ACM*, 41, 3 (March 1998), 105-108.

[18] Radermacher, A., and Walia, G. S. 2011. Investigating the effective implementation of pair programming: an empirical investigation. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (SIGCSE '11). ACM, New York, NY, USA, 655-660.

[19] Radermacher, A., and Walia, G. S. 2011. Investigating student-instructor interactions when using pair programming: an empirical study. To be published in *24th Conference on Software Engineering Education & Training* (CSEET '11). IEEE Computer Society, Washington, DC, USA.

[20] Radermacher, A. Walia, G. S., Myronovych, O., Abufardeh, S., and Rummelt, R. 2010. *Investigating the use of pair programming at North Dakota State University: a family of empirical studies*. Technical report, the department of computer science, North Dakota State University, http://cs.ndsu.edu/research/reports.

[21] Sabin, R. E., and Sabin E. P. 1994. Collaborative learning in an introductory computer science course. In *Proceedings of the 25th SIGCSE symposium on computer science education* (SIGCSE '94). ACM, New York, NY, USA, 304-308.

[22] Srikanth, H., Williams, L., Wiebe, E., Miller, C., and Balik, S. 2004. On Pair Rotation in the Computer Science Course. In *Proceedings of the 17th Conference on Software Engineering Education and Training* (CSEET '04). IEEE Computer Society, Washington, DC, USA, 144-149.

[23] Simon, Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D., and Tutty, J. 2006. Predictors of success in a first programming course. In *Proceedings of the 8th Austalian conference on Computing education - Volume 52* (ACE '06), Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 189-196.

[24] Williams, L., Kessler, R. R., Cunningham, W., and Jeffries, R. 2000. Strengthening the case for pair programming. *IEEE Software.* 17, 4 (July 2000), 19-25.

[25] Williams, L., Wiebe, E., Yang, K., Ferzli, M., and Miller, C. 2002. In support of pair programming in the introductory computer science course. In *Computer Science Education*, 12, 3 (September 2002), 197-212.

[26] Williams, L., Layman, L., Osborne, J., and Katira, N. 2006. Examining the Compatibility of Student Pair Programmers. In *Proceedings of the conference on AGILE 2006* (AGILE '06). IEEE Computer Society, Washington, DC, USA, 411-420.

[27] Williams, L. 2007. Lessons learned from seven years of pair programming at North Carolina State University. *SIGCSE Bull.* 39, 4 (December 2007), 79-83.

[28] Williams, L., McCrickard, D. S., Layman, L., and Hussein, K. 2008. Eleven Guidelines for Implementing Pair Programming in the Classroom. In *Proceedings of the Agile 2008* (AGILE '08). IEEE Computer Society, Washington, DC, USA, 445-452.

Programming at Different Levels: A Teaching Module for Undergraduate Computer Architecture Course

Xuejun Liang, Loretta A. Moore, and Jacqueline Jackson
Department of Computer Science, Jackson State University, Jackson, MS, USA

Abstract - *Computers can be programmed by using the high-level programming languages, such as C, C++, Java, Python, etc., and the low-level assembly programming languages, such as MIPS, IA-32, Accumulator, etc.. In this teaching module, we will use some simple computing examples to illustrate how to solve these same problems by using different computer programming languages. This work will expose students with different computer architectures and programming languages. It will show the similarities and differences among these architectures and programming languages. So, it will help students to get a better understanding of fundamentals of computer architectures and programming languages, and to enhance their problem solving skills with using computers.*

Keywords: Computer Architecture, Assembly Language, High-Level Programming Language, Problem Solving

1 Introduction

Computing with using computers can be carried out at different levels. At the application level, computer users can use any computer software for their application. For example, users can use the Microsoft Office Excel to organize and compute their data. At the high-level programming language level, computer programmers can use high-level programming languages, such as C, C++, Java, Python, etc., to write a computer program to solve their problems. At the low-level programming language level, programmers can use different machine assembly programming languages, such as MIPS, IA-32, Accumulator, etc., to write a computer program for the corresponding computer. It is quite difficult for a new computer science student to understand the similarities and differences among high-level programming languages and among low-level programming languages, and relationship between high-level and low-level programming languages. But, the above knowledge is important for students to have a deeper understanding of computer systems, to write correct and efficient computer programs, and to debug computer codes.

In this teaching module, we will use some simple examples to illustrate how to solve these computational problems using different means with computers. This work will help students to have a better understanding of fundamentals of computer architectures and programming languages. More importantly, this work will show how to design and implement a computer algorithm for a particular computer architecture or using a particular programming language. Therefore, it is highly expected that students will enhance their problem solving ability using computers significantly after they complete this teaching module.

In this teaching module, three simple computing problems are studied: (1) evaluate arithmetic expressions, (2) sort an array of integers, and (3) get an array of integer from the keyboard, sort the array, and display the sorted array on the screen. Three high-level programming languages are utilized: (1) Python, (2) Java, and (3) C++ (and C). Four computer architectures (assembly languages) are considered: (1) MIPS, (2) IA-32, (3) Accumulator-based machine, and (4) Stack-based machine. Eclipse is used for both C++ and Java programming. Real assembly languages are used for MIPS and IA-32 architectures, while the accumulator-based machine and stacked-based machine are simulated using Python, Java, and C++ for evaluating arithmetic expressions.

In the rest of the paper, the brief instructions and references for software download and installation are given first in Section 2. So, students can setup their computer for compiling and running the given codes and their own codes for the study. Then, the three computing problems and the solutions are discussed in Sections 3, 4, and 5, respectively. Finally, the conclusion and future work are given in Section 6.

2 Software Download and Installation

The software packages listed below are needed for running programs discussed in this paper. They all have been installed and tested on the Windows 7 platform.

2.1. **Python**: Download Python 2.7 release for Windows from http://www.python.org/download/releases/2.7/ [1] and select Windows x86 MSI Installer (2.7). Then run the downloaded program Python-2.7 for the installation.

2.2. **SPIM Simulator**: This simulator will be used for the MIPS assembly language programming. Download it from http://sourceforge.net/projects/spimsimulator/files/ [2] and select PCSpim_9.1.9.zip. Unzip it and run the setup program for the installation.

2.3. **MinGW**: (Minimal GNU for Windows): MinGW will provide a native Windows port of the GNU Compiler Collection (GCC), which will be used for linking IA-32 object code with C library I/O functions. Download MinGW from http://sourceforge.net/projects/mingw/files/ and click Installer and then select mingw-get-setup.exe. Then run the setup program for the installation.

2.4. **NASM**: (the Netwide Assembler): It is an assembler targeting the Intel x86 series of processors. It can get object codes from IA-32 assembly codes. Download NASM from http://www.nasm.us/pub/nasm/releasebuilds/2.10.09/win32/ and select nasm-2.10.09-win32.zip. Unzip it into a directory (for example, C:\NASM) where you want to install NASM. Then add the directory (for example, C:\NASM\nasm-2.10.09) where the nasm executable is stored into the Path environment variable.

To add C:\NASM\nasm-2.10.09 into the Path environment variable in Windows 7, from the Window's Start, select in turn the following: Control Panel, System and Security, System, Advanced system settings. Then, the System Properties dialogue window shows up, click the Environment Variables button. Under System variables list, find and select the variable Path, and click the Edit... button. Then type "C:\NASM\nasm-2.10.09;" at the end of the variable value. When you are done, click OK.

To compile and run x86 assembly language program, you will type `cmd` in Window's search programs and files textbox to get a command line window first. Then compile and link your program, say, exprX86.asm, by typing

```
>nasm -f win32 exprX86.asm
>gcc exprX86.obj -o exprX86
```

Finally, run the program by typing

```
>exprX86
```

2.5. **Eclipse IDE for Java Developers**: Download and unzip the file eclipse-java-indigo-SR2-win32-x86_64.zip from http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/SR2/eclipse-java-indigo-SR2-win32-x86_64.zip. Then, create a shortcut of eclipse.exe on your Desktop.

2.6. **Eclipse IDE for C++ Developers**: Download and unzip eclipse-cpp-indigo-SR2-incubation-win32-x86_64.zip from http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/SR2/eclipse-cpp-indigo-SR2-incubation-win32-x86_64.zip. Then, create a shortcut of eclipse.exe on your Desktop.

3 Evaluate Arithmetic Expressions

Consider the following arithmetic expression:

$$d = (a+b) * (c-b) + (b-c) + a \qquad (1)$$

Assume initial values a=32, b=45, and c=23. Now, we will evaluate this expression using the following programming languages and architectures.

3.1. **Python**: Figure 1 lists the Python code to evaluate (1).

```
a = 32
b = 45
c = 23
d = (a+b) * (c-b) + (b-c) + a
print d
```

Figure 1: Python Code to Evaluate the Expression (1)

3.2. **Java**: Figure 2 lists the Java code to evaluate (1).

```
public class ComputeExpression {
  public static void main(String[] args) {
    int a = 32;
    int b = 45;
    int c = 23;
    int d = (a+b)*(c-b)+(b-c)+a;
    System.out.println("The answer is  " + d);
  }
}
```

Figure 2: Java Code to Evaluate the Expression (1)

3.3. **C++**: Figure 3 lists the C++ code to evaluate (1).

```
#include <iostream>
int main() {
  int a = 32;
  int b = 45;
  int c = 23;
  int d = (a+b)*(c-b)+(b-c)+a;
  std:: cout << "The answer is " << d << endl;
  return 0;
}
```

Figure 3: C++ Code to Evaluate the Expression (1)

3.4. **MIPS**: Figure 4 lists the MIPS code to evaluate (1), where a, b, c, and d are assumed to be stored in memory locations A, B, C, and D, respectively. Note that the result display is omitted. The SPIM provides system calls to display data in registers or character strings stored in memory. From the code, we can see that MIPS is a load/store architecture (or register-to-register arithmetic) and there are three operands per ALU instructions.

```
  .text                # text segment
  .globl main
main:
  lw   $t0, A          # $t0 = a
  lw   $t1, B          # $t1 = b
  lw   $t2, C          # $t2 = c
  sub  $t3, $t2, $t1   # $t3 = c-b
  sub  $t4, $t0, $t3   # $t4 = a-(c-b) = (b-c)+a
  add  $t5, $t0, $t1   # $t5 = a+b
  mul  $t6, $t5, $t3   # $t6 = (a+b)*(c-b)
  add  $t7, $t6, $t4   # $t7 = (a+b(c-b)+(b-c)+a
  sw   $t7, D
  ......
  li  $v0,10
  syscall              # au revoir...

  .data                # data segment
A:    .word 32
B:    .word 45
C:    .word 23
D:    .word 0
```

Figure 4: MIPS Code to Evaluate the Expression (1)

3.5. **IA-32**: Figure 5 lists IA-32 code to evaluate (1), where a, b, c, and d are assumed to be stored in memory locations A, B, C, and D, respectively. Note that the result display is omitted. The NASM allows to use C library function `_printf` to display data. Programmers for IA-32 can simply call it in the assembly code after pushing its actual parameters onto the stack. From the code, we can see that IA-32 is not a load/store architecture (or register-to-register arithmetic) and there are two operands per ALU instructions. So, the output operand of ALU instructions is the same with the first input operand, and only the second input operand can be a memory location.

```
   global  _main
   extern  _printf

   section .text   ; Text section
 _main:
   mov  eax, [A]   ; eax = a;
   mov  ebx, [B]   ; ebx = b;
   add  eax, ebx   ; eax = a+b;
   sub  ebx, [C]   ; ebx = b-c;
   imul ebx        ; edx:eax = (a+b)*(b-c);
   add  ebx, [A]   ; ebx = (b-c)+a;
   sub  ebx, eax   ; ebx = (a+b)*(c-b)+(b-c)+a;
   mov  [D], ebx   ; [D] = (a+b)*(c-b)+(b-c)+a;
   ......
   ret

   section .data   ; Data section
 A:      dd  32
 B:      dd  45
 C:      dd  23
 D:      dd  0
```

Figure 5: IA-32 Code to Evaluate the Expression (1)

3.6. **Accumulator**: Assume that accumulator-based machine has the following five instructions (functions), where *acc* is the accumulator, as shown in Table 1. Then, Figure 6 lists the accumulator code to evaluate (1). Note that the accumulator machine has only one operand and the rest of operands is *acc*.

Table 1: Instructions of Accumulator Machine

Instruction	Meaning
ld(A)	acc = memory[A]
add(A)	acc = acc + memory[A]
mul(A)	acc = acc * memory[A]
sub(A)	acc = acc - memory[A]
st(A)	memory[A] = acc

```
# Accumulator code
ld(a)    #acc = a
add(b)   #acc = a+b
st(d)    #d = a+b
ld(c)    #acc = c
sub(b)   #acc = c-b
mul(d)   #acc = (a+b)*(c-b)
add(b)   #acc = (a+b)*(c-b)+b
sub(c)   #acc = (a+b)*(c-b)+(b-c)
add(a)   #acc = (a+b)*(c-b) +(b-c)+a
st(d)    #d = (a+b)*(c-b) + (b-c) + a
```

Figure 6: Accumulator Code to Evaluate the Expression (1)

In order to run the code shown in Figure 6, we can define and use the instructions (or functions) in Table 1 using high-level programming languages and then run the code in simulation.

3.6.1. **Python Simulation**: Figure 7 lists the Python code to simulate the accumulator machine and to evaluate the expression (1).

```
acc = 0

Def ld(A):
  global acc
  acc = A
def add(A):
  global acc
  acc = acc + A
def mul(A):
  global acc
  acc = acc * A
def sub(A):
  global acc
  acc = acc - A
def st(A):
  global acc
  A[0] = acc

a = 32
b = 45
c = 23
d = [0]

# Accumulator code

print d
```

Figure 7: Python Code to Simulate the Accumulator Machine

From the code in Figure 7, we can see that *acc* is implemented as a global variable and the array type parameter is used for passing the result back from the function `st(A)` because the Python function uses the call-by-value only.

3.6.2. **Java Simulation**: Figure 8 lists Java code to simulate the accumulator machine and to evaluate the expression (1).

```
public class ExprAccumulator {
  static int acc;

  static void ld(int A){
    acc = A;
  }
  static void add(int A){
    acc = acc + A;
  }
  static void sub(int A){
    acc = acc - A;
  }
  static void mul(int A){
    acc = acc * A;
  }
  static void st(int [] A){
    A[0] = acc;
  }

  public static void main(String[] args) {
    int a = 32;
    int b = 45;
    int c = 23;
    int[] d = new int[1];

    //# Accumulator code

    System.out.println("The answer: " + d[0]);
  }
}
```

Figure 8: Java Code to Simulate the Accumulator Machine

From the code in Figure 8, we can see that *acc* and all the functions are static. The array type parameter is used for passing the result back from the function `st(int[]A)` because the Java function uses the call-by-value only.

3.6.3. **C++ Simulation**: Figure 9 lists C++ code to simulate the accumulator machine and to evaluate the expression (1).

```
#include <iostream>
using namespace std;

int acc;

void ld(int A){
  acc = A;
}
void add(int A){
  acc = acc + A;
}
void mul(int A){
  acc = acc * A;
}
void sub(int A){
  acc = acc - A;
}
void st(int &A){
  A = acc;
}

int main() {
  int a = 32;
  int b = 45;
  int c = 23;
  int d = 0;

  //# Accumulator code

  std::cout << "The answer: " << d << endl;
  return 0;
}
```

Figure 9: C++ Code to Simulate the Accumulator Machine

From the code in Figure 9, we can see that *acc* is implemented as a global variable. The C++ function `st(int &A)` now uses the call-by-reference for passing the result.

3.7. **Stack-based**: Assume that the stack-based machine has the following five instructions (functions) as shown in Table 2. Then, Figure 10 lists the stack-based code to evaluate (1).

Table 2: Instructions of Stack-based Machine

Instruction	Meaning
push(A)	tos++; S[tos] = memory[A]
pop(A)	memory[A] = S[tos]; tos--
add	S[tos-1] = S[tos-1]+S[tos]; tos--
sub	S[tos-1] = S[tos-1]-S[tos]; tos--
mul	S[tos-1] = S[tos-1]*S[tos]; tos--

```
#Stack-based code
push(a)
push(b)
add()      # a+b
push(c)
push(b)
sub()      # c-b
mul()      # (a+b)*(c-b)
push(b)
push(c)
sub()      # b-c
add()      # (a+b)*(c-b)+(b-c)
push(a)
add()      # (a+b)*(c-b)+(b-c)+a
pop(d)
```

Figure 10: Stack-Based Code to Evaluate the Expression (1)

Note that all operands of ALU instructions are located in the operand stack. Therefore, ALU instruction has zero operand in the stack-based machine. The code shown in Figure 10 can be obtained from the postfix notation of Expression (1), which is

$$d = a\ b + c\ b - * \ b\ c - + \ a + \qquad (2)$$

In order to run the code shown in Figure 10, we can define and use the instructions (or functions) in Table 2 using high-level programming languages and then run the code in simulation. We can use an array to simulate the operand stack as shown in Table 2. At the mean time, we can use the stack data structure in high-level language or library to simulate the operand stack. These codes are omitted here for the space.

4 Sorting Array

Consider sorting the integer array: 68, 86, 65, 75, 67, 76, 62, 79, 98, 67 in an ascending order.

4.1 **Using Built-in Sorting Function**: Figures 11, 12 and 13 list the Python code, Java code, and C++ code that use their built-in array sorting function to sort the array, respectively. It can be noticed that the codes are all quite simple. The Python code are only three lines! Note that printing an integer array needs to have a loop structure in C++, but the loop is not needed in Python and Java.

```
a = [68,86,65,75,67,76,62,79,98,67]
a.sort()
print a
```

Figure 11: Python Code to Sort an Array

```
import java.util.Arrays;

public class Sorting {
  public static void main(String[] args) {
    int[]a ={68,86,65,75,67,76,62,79,98,67};
    Arrays.sort(a);

    //print results
    System.out.println(Arrays.toString(a));
  }
}
```

Figure 12: Java Code to Sort an Array

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
  const int len = 10;
  int a[len]={68,86,65,75,67,76,62,79,98,67};
  sort(a, a+len);

  //print results
  for (int i=0; i< len; i++)
    cout << a[i] << " ";
  cout << endl;

  return 0;
}
```

Figure 13: C++ Code to Sort an Array

4.2 **Using the Bubble Sort Algorithm**: Consider putting the elements in a vertical array. The bubble sort algorithm will scan the array elements from bottom to up repeatedly until the array is sorted. In each scan, every adjacent pair of elements are compared. They are swapped if they are not in right order. So, after the first scan, the first element in the array is already in its right place. Similarly, the second scan will make the second element in the array to be in its right place, and so on.

4.2.1. **Python, Java, and C++**: Figure 14 lists Python code that uses the bubble sort algorithm to sort the array. As you can guess, the bubble sort implementations in Java and C++ will be identical, and they are different from Python only in syntax. Java and C++ use { } to include a block of statements, while Python use : and the indentation to indicate a block of statements. Variables should have an explicit data type in Java and C++, but not in Python. Therefore, Java and C++ codes that use the bubble sort algorithm are omitted here.

```
a = [68,86,65,75,67,76,62,79,98,67]
len = len(a)

current = 0;
while (current < len - 1):
  index = len-1
  while (index > current):
    if (a[index] < a[index-1]):
      tmp = a[index]
      a[index] = a[index-1]
      a[index-1] = tmp
    index = index - 1
  current = current + 1

print a
```

Figure 14: Python Code: Bubble Sort

4.2.2. **MIPS and IA-32**: After the bubble sort algorithm has been implemented using a high-level programming language, a better way to implement it in an assembly language again is to consider how to translate branch and loop statements from high-level languages to low-level languages. In Table 3, the second column lists corresponding constructs in assembly languages for branches and loops in high-level language.

Table 3: Branch and Loop in Assembly Language

High Level Language	Assembly Language

if(x<y)
  A;
C;
	if(x>=y) goto c
	A
	c: C

if(x<y)
  A;
else
  B;
C;
	if(x>=y) goto b
	A
	goto c
	b: B
	c: C

While(x<y)
  A;
C;
	a: if(x >= y)goto c
	A
	goto a
	c: C

Figures 15 and 16 list MIPS code and IA-32 code that both use bubble sort algorithm to sort the array, respectively. One key point in assembly language programming is how to access array elements, i.e. how to compute the memory addresses of array elements. Simply speaking, the address of an array element can be computed by adding the array base address and the array element offset, which is the product of the array element index and the element size. As the element size in the code is four, the element offset should be multiple of four.

```
  .text
  .globl main
main:
  move $t0, $0       # $t0 = current
  lw   $t1, count    # $t1 = 10
  sll  $t1, $t1, 2   # $t1 = 40
  addi $t1, $t1, -4  # $t1 = 36
loop1:
  beq  $t0, $t1, done
  move $t2, $t1      # $t2 = index
loop2:
  beq  $t2, $t0, next
  addi $t3, $t2, -4
  lw   $t4, A($t2)
  lw   $t5, A($t3)
  bge  $t4, $t5, continue
  sw   $t5, A($t2)
  sw   $t4, A($t3)
continue:
  addi $t2, $t2, -4
  b    loop2
next:
  addi $t0, $t0, 4
  b    loop1
done:
  li   $v0,10
  syscall            # au revoir...

  .data
A:     .word 68,86,65,75,67,76,62,79,98,67
count: .word 10
```

Figure 15: MIPS Code: Bubble Sort

```
  global  _main
  section .text
_main:
  sub  edi, edi      ; edi = current
  mov  ecx, [count]  ; ecx = 10
  shl  ecx, 2        ; ecx = 40
  sub  ecx, 4        ; ecx = 36
loop1:
  cmp  edi, ecx
  je   done
  mov  eax, ecx      ; eax = index
loop2:
  cmp  eax, edi
  je   next
  mov  ebx, [eax+A]
  mov  edx, [eax+A-4]
  cmp  ebx, edx
  jge  continue
  mov  [eax+A], edx
  mov  [eax+A-4], ebx

continue:
  sub  eax, 4
  jmp  loop2
next:
  add  edi, 4
  jmp  loop1
done:
  ret

  section .data
A:      dd  68,86,65,75,67,76,62,79,98,67
count:  dd  10
```

Figure 16: IA-32 Code: Bubble Sort

From Figures 15 and 16, it can be seen that both codes have the same program structure with the Python code. Registers `$t0`, `$t1`, and `$t2` in MIPS code and registers `edi`, `ecx`, and `eax` in IA-32 code have the values of `current*4`, `(len-1)*4`, and `index*4`, respectively, where `current`, `len-1`, and `index` are used in the Python code.

5 Problem Decomposition

In this section, we will consider the problem: get an array of integers from the keyboard, sort the array, and display the sorted array on the screen. This task is composed by three subtasks: (1) input an array, (2) sort an array, and (3) print an array. We want to use a subroutine for array sorting, while the input and the output will remain in the main routine.

5.1 **Python, Java, C++, and C**: Utilizing a subroutine in high-level language is easy as we see in subsections 3.6.1, 3.6.2, and 3.6.3. Printing an integer array in high-level language is also simple. We will only show how to accept an integer array from the keyboard. These are shown in Figures 17-20.

```
a = []
length= int(raw_input("Enter the array length: "))
for i in range(0, length):
  if( i = 0):
    a.append(int(raw_input(Enter your integer array)))
  else
    a.append(int(raw_input()))
```

Figure 17: Python Code: Accept Input Array

```
public class SortBubble {
  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter the array length:  ");
    int length = in.nextInt();
    int [] a = new int[length];
    System.out.print("Enter your integer array:  ");
    for(int i = 0; i < length; i++)
      a[i] = in.nextInt();
    }
}
```

Figure 18: Java Code: Accept Input Array

```
int main() {
  int length;
  cout << "Enter the array length: " << endl;
  cin >> length;
  int a[length];
  cout << "Enter your integer array: " << endl;
  for(int i = 0; i < length; i++)
    cin >> a[i];
  return 0;
}
```

Figure 19: C++ Code: Accept Input Array

```
int main() {
  int length;
  printf("Enter the array length: " );
  scanf("%d", length);
  int a[length];
  printf( "Enter your integer array:" );
  for(int i = 0; i < length; i++)
    scanf(%d", a[i]);
  return 0;
}
```

Figure 20: C Code: Accept Input Array

Note that Figure 20 shows the C code that uses `printf` and `scanf` functions to do the I/O. We will need this knowledge to do the I/O for IA-32 assembly programs with using NASM.

5.2 **MIPS and IA-32**: Firstly, how to accept input array from the keyboard and print it on the screen is shown in Figure 21 and 22 for MIPS and IA-32, respectively. Both codes get the array length first and then get array elements using a loop. Both codes also print the array using a loop. The array length is stored at `count` and the array is stored starting from `A`. In MIPS code in Figure 21, the system calls are used for both read an integer from keyboard and print an integer on screen. In IA-32 code in Figure 22, `_scanf` is called for reading and `_printf` is called for printing. Note that before calling them, we need to push the arguments onto the stack.

```
  .text
  .globl main
main:
  #Read count
  li    $v0, 5
  syscall           # read an integer
  sw    $v0, count

  # Read integer array
  move  $t0, $0
  lw    $t1, count
  sll   $t1, $t1, 2
loop1:
  beq   $t0, $t1, done1
  li    $v0,5
  syscall           # read an integer
  sw    $v0, A($t0)
  addi  $t0, $t0, 4
  b     loop1
done1:

  # print the integer array
  move  $t0, $0
  lw    $t1, count
  sll   $t1, $t1, 2
loop2:
  beq   $t0, $t1, done2
  lw    $a0, A($t0)
  li    $v0,1
  syscall            # print an integer
  la    $a0, sp
  li    $v0,4
  syscall            # print the space
  addi  $t0, $t0, 4
  b loop2
done2:

  li    $v0,10
  syscall            # au revoir...

  .data
A:     .word 0:20
count: .word 0
sp:    .asciiz " "
```

Figure 21: MIPS Code: Accept Input Array and Print It

Secondly, as shown in Figure 23 and Figure 24, the subroutine `bubbleSort` is designed to have two input arguments. The first one is the array starting address and is passed to the subroutine via `$a0` and via `edi` for MIPS and IA-32, respectively. The second one is the array length and is passed via `$a1` and via `ecs` for MIPS and IA-32, respectively. The sorted array will remain in the same memory locations.

```
  global  _main
  extern  _printf
  extern _scanf

  section .text
_main:
  ;read count
  push  count
  push  rfmat
  call  _scanf       ;read integer
  add   esp, 8

  ;read integer array
  mov   ecx, [count];
  mov   edi, A
read:
  push  ecx
  push  edi
  push  rfmat
  call  _scanf       ;read integer
  add   esp, 8
  add   edi, 4
  pop   ecx
  loop  read

  ;print integer array
  mov   edi, A
  mov ecx, [count]
print:
  push  ecx
  mov   ebx, [edi]
  push  ebx
  push  wfmat
  call  _printf ;print integer
  add   esp, 8
  add   edi, 4
  pop   ecx
  loop  print
  ret

  .section .bss
A:      resd 20
count:  resd 1

  .section .data
rfmat:  db  '%d', 0
wfmat:  db  '%6d', 0
end:    db  10, 0
```

Figure 22: IA-32 Code: Accept Input Array and Print It

```
  .text
  .globl main
main:
  # Read count
  # Read integer array and store at A

  la    $a0, A
  lw    $a1, count

  jal   bubbleSort

  # print the result

  li $v0,10
  syscall           # au revoir...

bubbleSort:
  ......
  jr   $ra
```

Figure 23: MIPS Code with Subroutine Call

```
  global  _main
  extern  _printf
  extern  _scanf

  section  .text
_main:
  ;Read count
  ;Read integer array and store at A

  mov   edi, A
  mov   ecx, [count]

  call  bubbleSort

  ;Print the sorted integer array:
  ret

bubbleSort:
  ......
  ret
```

Figure 24: IA-32 Code with Subroutine Call

Note that Figure 23 and 24 show the whole program skeletons. the subroutine bubbleSort codes are omitted here in Figure 23 and 24 because they have almost identical codes with those shown in Figure 15 and 16. The codes of reading an input integer array and printing the output array can be found from Figure 21 and 22.

6 Conclusion and Future Work

The teaching module was taught last semester and students were given a related assignment as bonus. Most students are very interested in this material and like the assignment. Some future work can be as follows.

- Illustrate the stack-based machine by using the Java byte code (or Java Virtual Machine) directly.
- Illustrate implementations of basic data type such as two-dimensional array and class at assembly language level.
- Provide some applications that can be more effectively and simply solved by using assembly languages than high-level programming languages.
- Multi-core processor programming and architecture.
- GPU (graphics processing unit) programming model and architecture.

7 References

[1] Python 2.7 Release, Available at http://www.python.org/download/releases/2.7/

[2] Sourceforge, *Spim MIPS simulator*, Available at http://sourceforge.net/projects/spimsimulator/files/

[3] Sourceforge, *MinGW-Minimal GNU for Windows*, Available at http://sourceforge.net/projects/mingw/files/

SESSION

TOOLS AND SYSTEMS + ATTENDANCE TRACKING, CLASS MANAGEMENT SYSTEMS, TESTING AND QUIZZ SYSTEMS, INSTRUCTIONAL DESIGN AND RELATED ISSUES

Chair(s)

TBA

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

WUTexter: A Classroom Interaction Tool For Anybody Who Can Text

Benjamin Murray, Hunter LaTourette[1], and Ron K. Cytron[1]
[1]Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, USA

Abstract— *The classroom experience is enhanced for both teacher and student alike when feedback mechanisms are in place to measure the students' engagement. Specialized devices such as the* `iClicker` *have been developed along these lines, but these devices have limited range and use. This paper describes the design and implementation of a device that can be used within reach of the Internet or cell tower, allowing students to interact with the instructor in the classroom by sending simple text messages. The application currently accommodates polls, student-supplied questions, and a mechanism to express confusion or boredom. We provide some early anecdotal experiences using this device, articulate plans for the future, and describe how others can use this technology in their own settings.*

Keywords: classroom interaction, texting

1. Introduction

The context for the work described in this paper began with our department's decision to purchase 50 `iClicker` [1] devices for use in our department's courses. While many schools require or encourage (through extra credit) each student to purchase his or her own `iClicker`, we opted for a departmental basket containing 50 of these devices. An example of such a device currently in production and for sale is shown in Figure 1. An instructor would bring these devices to class, distribute them randomly among the students present, and construe the results of their use as statistically valid random sampling of the (perhaps larger) classroom population.

These plans did not account for the combined battery weight of 50 such devices. With two batteries per device, the basket of `iClicker`s was too heavy to carry easily across campus. Upon hearing this, we began to consider how students could participate in feedback such as that sought by the `iClicker`, but by using technology already in their possession. It occurred to us that all students we knew could *text* using devices that connect to the cell networks. Most students are exceptionally adept at texting, owing to the frequency they communicate with friends and family using texting. Many, if not most, of their devices can also connect to the Internet.

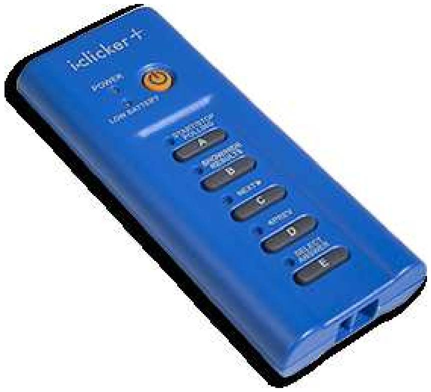
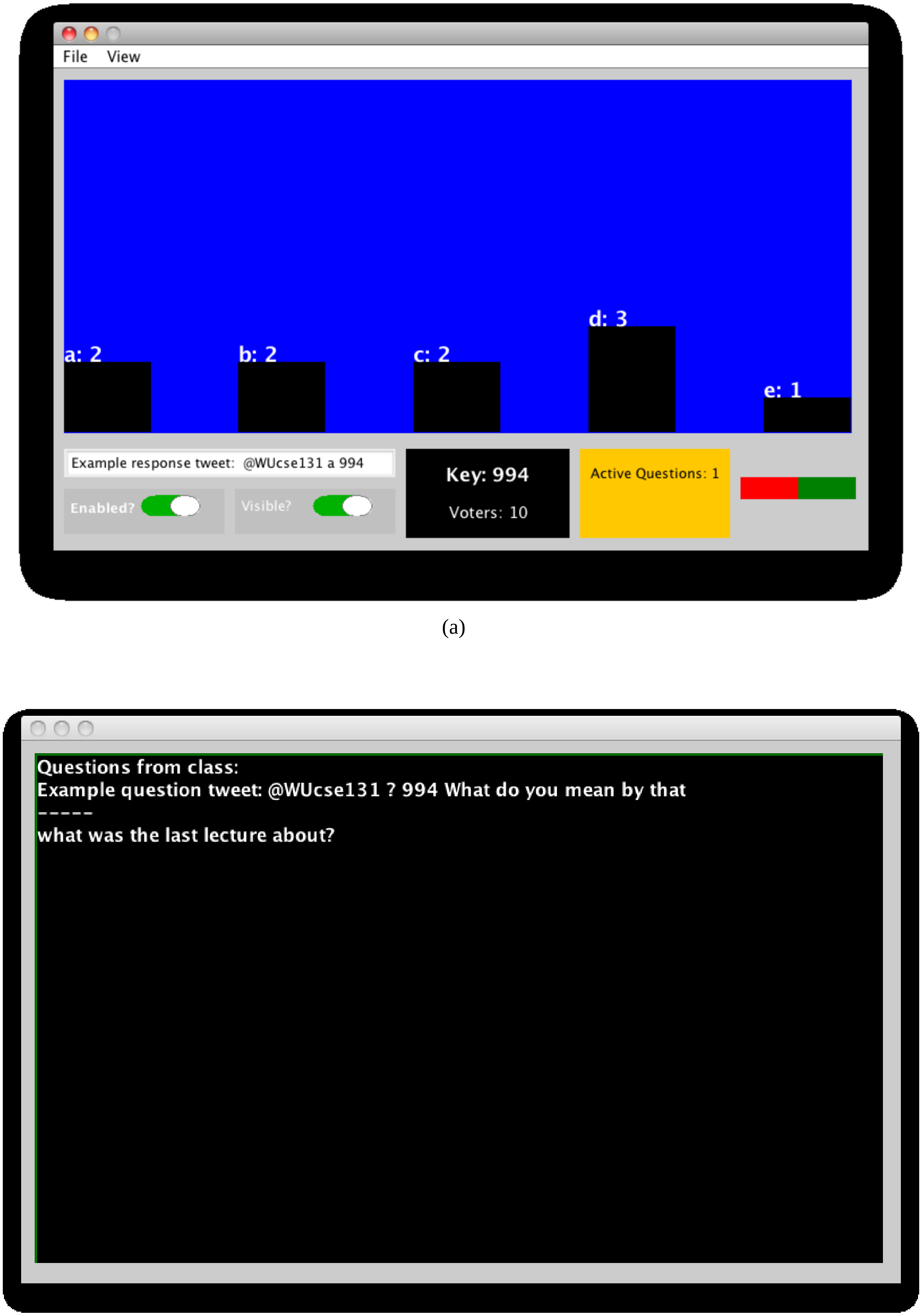


Figure 1: An `iClicker` device. Five buttons are provided for its user to respond to polls. From `http://www.iclicker.com`.

While our initial motivation for the work we present in this paper was to free our instructors from carrying the `iClicker`s to class, we arrived at the following arguments in favor of our approach:

- `iClicker`s require a base receiver for operation. This base receiver is purchased separately from the `iClicker`s, and it must be brought to class and connected to a computer to receive and tabulate students' responses.
- `iClicker`s are limited in transmission range. Due to FCC regulations, the `iClicker` devices are limited in the strength of signal they are allowed to emit. The devices and base station work within a room, but they cannot work for students situated remotely from the base receiver.
- `iClicker` devices are very limited in the form of response. Referring to Figure 1, a student presses one of five buttoms to register response. In particular, these devices lack the ability to *initiate* interaction with the instructor, to pose a question or to indicate confusion or boredom.

While the design and implementation are described in greater detail below, we begin with a quick description of the features of `WUTexter`. The application launches into the two panels shown in Figure 2. The main panel (a) controls the application and displays almost all of the information. The other panel (b) serves to display only the text of questions posed by the participants. This separation into two panels is convenient in terms of how the information



```
Example response tweet:  @WUcse131 a 994
Enabled? ●
Visible? ●
Key: 994
Voters: 10
Active Questions: 1
```

(a)

```
Questions from class:
Example question tweet: @WUcse131 ? 994 What do you mean by that
-----
what was the last lecture about?
```

(b)

Figure 2: (a) The primary `WUTexter` control panel and display. (b) Participants' questions are displayed in a separate panel.

is displayed in the classroom. An instructor can choose to place the students' questions panel on a private display, while still showing the results of polls on a screen seen by the participants. The components of the display shown in Figure 2(a) are as follows:

- The largest portion of the window is dedicated to showing the results of polling. Each vertical bar indicates by its height the fraction of participants who have chosen that particular response. The example shown here represents 10 participants' responses to a multiple-choice poll offering 5 possible responses. As shown, responses `a`, `b`, and `c` have each received two votes apiece. Response `d` earned 3 votes and response `e` earned only 1 vote.

  This portion of our tool replicates the functionality found in the `iClicker`. A small difference is that the number of available choices can be configured when the tool is launched, with the default of 5 to match the `iClicker`.
- Continuing down the screen we see an example tweet to help students model their responses:

  ```
  Example response tweet:  @WUcse131 a 994
  ```

  The response tweet contains three pieces of information:
  - The `twitter` account name for the class, appended to an at-sign (`@`).
  - The student's desired response (`a`, `b`, etc.).
  - A 3-digit secret key for this `WUTexter` session (Section 3.5). This key makes it difficult for those not privy to the lecture to participate in the `WUTexter` session. The key can be changed at will by the instructor.
- Shown below are two sliders that control whether the poll is enabled (open) or not (closed), and whether the polling results are dynamically visible to the audience.
- Continuing to the right, a black box at the bottom shows the current secret key (994 in our example) and tabulates the number of participants in the current poll.
- Continuing to the right, a box indicates by its background color whether questions have been asked by the students. The background remains a bright color until the instructor clicks on the box to reset the tally to 0. At that point, presumably the instructor has addressed the posted questions or has chosen to ignore them.

  A question is posed when a participant substitutes a `?` for a poll choice such as `a`, in the position where the poll choice would normally appear. As before, the secret key must be the next token supplied on the response. The remaining text is interpreted as the participant's question and is posted on the panel shown in Figure 2(b).
- The final component to the right is a red and green bar that reflects the participants' view that the material should be delivered more quickly or slowly. A student does this by furnishing a `+` sign instead of a letter in the model response. Similarly, a `-` sign requests that lecture slow down. This is described in greater detail in Section 4.3.
- The `File` and `View` menus allow instructors to access other functionality, such as resetting the key, zeroing the tabulations, and exporting a log file of those who have participated in the session.

While in-depth studies of `WUTexter` have not been conducted, we report on the tool in this paper so as to make it available for more widespread use.

2. Related work

The use of mobile devices in the hands of students and instructors is currently a topic of great conversation in academia [2]. Not surprisingly, computer science courses are among the first to adopt and adapt to such technology [3]. Tim Hickey and others [4] have considered student feedback to affect the pace of the classroom. They have also developed a system that allows peer-instruction during class, so that students can field and answer other students' questions.

3. Approach

Seeking a method by which any student, anywhere, could participate in an `iClicker`-like poll, it became clear that we would need to rely on the Internet or the cell phone network for communication. Broadly speaking, we required an *event channel* [5] that could relay messages from students to an application that could tabulate results, report students' questions, and keep track of the students' confusion or boredom.

Many social media sites allow messaging among their participants. However, privacy concerns typically limit such messaging to those who have explicitly allowed such activities. Moreover, we do not need these messages to persist, so it would help if the messages are easily deleted by the receiver (our application). `Facebook` surfaced early as a candidate, but students understandably protect their `facebook` presence. Also, `facebook` distinguishes messages sent between friends and non-friends. We could not require students to be friends with the class account. We soon turned to `twitter` for the following reasons:

- `Twitter` is primarily concerned with messaging, and other features are less prominent. For `facebook`, messaging is not its primary mission.
- It is easy to create `twitter` accounts that use the same `gmail` account. Thus, a professor could have separate accounts for each class without having to create a new email address for each one.
- The API seemed easy to use. An application could generate authentication keys, which can then be used by an application to create a `twitter` session.

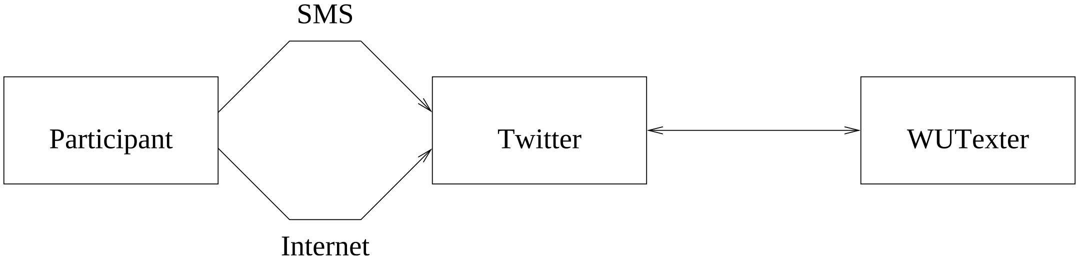


Figure 3: Software architecture of `WUTexter`.

- `Twitter` can use `SMS`, through which text messages of very simple syntax can simulate tweets. This puts our approach within reach of any student who has access to the Internet or who can text, even from a very simple phone.
- While `twitter` does limit any user to sending no more than 250 messages per hour, there is no such limit on the messages an account can receive.

Figure 3 shows the software architecture of `WUTexter` using `twitter` as an event channel. The participant reaches `twitter` using either the cell tower via SMS (Short Message Service) or over the Internet. To use SMS, the participant simply sends a text message to a special SMS "phone number" (40404 for `twitter`). The body of the text message is of the format described previously. If the participant has access to the Internet, perhaps using a data plan on a mobile device, then he or she can reach `twitter` in the usual way, by logging into `twitter` via a browser or by accessing his or her `twitter` account using an application.

Once the participant has tweeted a message, the `WUTexter` box in Figure 3 picks up such messages by listening for references to its account name. In Figure 2, the account shown is `WUcse131`. The case-spelling of the account name does not matter, but the capitalization serves to show how the account name was chosen: WU for Washington University and the rest of the account name is a course at the university.

Any `twitter` account can be used, but it is helpful to choose an account for a course whose name is easily remembered. The tweet message must mention the account name (recall `@WUcse131`), and the `WUTexter` application must access `twitter` by authenticating to the same account. A personal `twitter` account could be used, but we find it convenient to dedicate an account to a particular course. That same account is then used each semeter regardless of who teaches that course.

3.1 Instructor preparation

To use `WUTexter`, an instructor must decide on a `twitter` account to be used as the target of messages sent by the students. As described above, we establish a separate `twitter` account for each course using `WUTexter`. For example, our CSE131 course has the `twitter` account `wucse131`. Accounts named in this way should be readily available at `twitter`. Generally, the following steps are needed to set up such an account:

1) On the `twitter` home page, request an account with a suitable name.
2) Each `twitter` account needs an email address to which it is associated. If the instructor already has a `gmail` account (say, `professor@gmail.com`), then an email address of the form

   `professor+wucse131@gmail.com`

   can be used, because `gmail` disregards the portion at and after the `+` sign. Thus, confirmation (and any other desired) messages will be sent to the instructor's specified `gmail` account.
3) An instructor typically has no reason to see the email messages generated by `twitter`. The preferences section of a `twitter` account can be visited and email notifications can be disabled. In practice, this results in almost no messages at all being sent to the associated account.
4) Using the created and verified `twitter` account, log into `dev.twitter.com`. Here, you will generate an application tied to the account.
5) Finally, generate authorization keys so that the `WUTexter` application can log in under your new account to receive tweets from students. The text of these keys is saved in a configuration file used by `WUTexter`

3.2 Student preparation

For the student side of this application, a `twitter` account is needed, and the tweets of that account must be publicly visible (such is the default for a new `twitter` account). Students may decide to use their personal or otherwise extant `twitter` accounts, or they may decide to establish a new one specifically for use with `WUTexter` or for a given course. In terms of participating in `WUTexter` activity, there is no difference.

However, some instructors may elect to give credit based on participation, in which case the association of a twitter account with a student must become known so that credit could be given. WUTexter can save log files from its sessions, and the correspondence between a twitter account name and a particular student must be discovered at some point to award credit. In our courses, we typically keep this correspondence secret until the end of the semester, so that students can use WUTexter anonymously. We find that students are more likely to pose questions if their identity is secret. At the end of the semester, students can tweet their student ID at the class twitter account, so that credit can be awarded for participation if that is the instructor's policy.

3.3 Twitter as an event channel

Messages related to WUTexter are sent from student to the classroom tool via twitter. Recalling the form of a message, the following

```
@wucse131 c 377
```

sends a message to vote for option c using secret key 337. The message is intended for the wucse131 twitter account, and the at-sign is necessary in front of the account name so that twitter alerts the class account on behalf of this message.

Although a large number of students may be voting at once, each is using his or her own twitter account. Currently each such account holder is limited to some 250 messages per hour, which is ample to accommodate the intended usage here.

The receiver of these messages, the wucse131 account in the above example, cannot be held responsible for those *mentioning* the wucse131 account name in a tweet. There is thus no limit, from what we have seen, in terms of the number of messages the account can receive.

There are two ways in which messages can be sent:

- **tweeting** The examples have thus far used this mechanism. The account is mentioned using the at-sign notation, which causes the message to be sent to the account as a *status* update.
- **direct message** Twitter allows its users to send direct messages to each other. A student can send a direct message to the class account, providing that the class account is *following* the student's account. Thus, once per session, our in-classroom WUTexter tool responds to a status message by *following* the source of that message, thereby making sure that direct messages are possible.

Why use direct messages instead of tweets? There are two reasons:

- A direct message is private, and would not be displayed in a user's twitter feed.
- A given tweet can be sent only once. Consider a student participating in a poll, who initially feels a is the correct response. If that student changes his or her mind to b and then back to a, the last tweet to assert the response as a is redundant and is therefore suppressed by twitter. The last nonredundant message would the final say for such a student, so it would appear this student voted for b.

  A direct message can be sent twice, so that mechanism allows students to change their minds more frequently and fluidly.

In terms of ease of use, tweets are slightly shorter than direct messages. The improvements we have in mind for the student-client would automatically favor direct messages over tweets. This is discussed in Section 5.

3.4 Running the in-classroom server

There are two items necessary to run the server:

- A copy of the WUTexter jar file.
- A configuration file

The jar file contains the code as described below. The configuration file contains the authorization keys necessary for a twitter account to open a session with twitter. The server by default opens the file config.txt as its configuration file, but this can be changed when the server is started. It is probably most convenient simply to store a given class's configuration file as config.txt.

The WUTexter server that runs in the classroom is written entirely in Java, and uses the twitter-4j library to access the twitter API. The resulting code is packaged in a jar file so that it can be run in any classroom that supports Java. Double-clicking the jar file icon typically suffices to launch it. From a command line prompt, the following also runs the server:

```
java -jar WUTexter.jar
```

With the application written in Java, we have seen no problems at all running the server in classrooms across our campus. The server can be run on an in-classroom computer (Java is typically present) or on the instructor's laptop.

3.5 Secret key

Each WUTexter session creates a 3-digit key. Messages that are missing the key are ignored. The purpose of this key is to ensure that only those students who are participating in the class (locally or remotely) can send messages that are considered by the application. While the logical basis for the key is sound, the key requires extra work on the part of the student, and we return to this point in Section 5.

4. WUTexter modes and experience

The WUTexter tool has been used in several classes at Washington University, with enrollments from 30 to over 300. In rooms with multiple projectors, it is possible to have the display shown continuously on one screen while lecture materials are presented on the other screen. For rooms with a single projector, many laptops offer the ability to show different images on the laptop and the projected screen. In such cases, `WUTexter` could be running in the instructor's view, and dragged to the students' view as needed. As seen in Figure 2, there are various and concurrent ways in which `WUTexter` can be used, as described below.

4.1 Polling

In terms of its conceived design as a replacement for `iClicker`, the most apparent mode of operation for `WUTexter` is to offer students a mechanism for responding to a poll. The purposes for such polls include the following:

- How prepared are students for class? Did they read the assigned materials?
- How well are students following the lecture material thus far? Does some material need to be covered again?
- How well can students apply what they have learned in new settings?

Thanks to the `iClicker`'s widespread availability and use, others have considered ways in which they can be used for conducting polls [6]. The mechanism supplied here is identical to `iClicker`, except that students can participate remotely and use only their cell or smart phones to do so. The experience so far with polling has been positive, but we remark in Section 5 on improvements we can make client-side that would make this and other functionality easier for students to use.

4.2 Posing questions

Students typically have questions, and many of these students are reluctant to ask questions in class for a variety of reasons:

- They don't want to interrupt the lecture.
- They don't want to disclose their uncertainty, confusion, or misunderstanding.
- They may believe they are the only ones who misunderstand, so they are reluctant to take up class time to get an answer to their questions.

As a result of the above disincentives, many students *disconnect* with lecture at the point where they have a question but feel unable to pose the question to the instructor. This disconnection may occur because the given student's mind is preoccupied with the question, or it may be the case that because of the question, the student feels (perhaps hopelessly) lost.

While designing the `WUTexter` tool, we realized that we could accommodate information well beyond that of a poll and its responses. A channel was effectively open by which students could ask questions anonymously. The questions would appear on the screen, so that a professor need not stop mid-sentence. The professor can decide whether and in what manner to answer the question.

When deployed in a classroom, this feature seemed to earn the most attention and praise. `WUTexter` can be enabled as students are entering the class, some 10 minutes before class officially started, and students fell into the habit of sending in questions right away. This allows instructors to respond to common concerns before class started. Students reported that they really liked and appreciated the anonymity with which they could pose questions. With students having an essentially open forum to express themselves, it would have been possible for them to send annoying or inappropriate comments, but this never occurred.

4.3 Confuse-o-meter

There are occasions where a student would like lecture to speed up or slow down, without having a particular concern or question to pose. For such students, we offer the confuse-o-meter, which could have the following interpretations:

- Should the lecturer speed up, keep the current tempo, or slow down?
- Is the class generally fine, confused, or bored?

Students can send messages via `WUTexter` to cause the confuse-o-meter to deflect according to their mood. The display in Figure 2 is a horizontal green and red bar. With equal amounts of each color, the class is fine. As the bar becomes more green, the lecturer should increase the pace, and as the bar becomes more red, the lecturer should slow down. The bar has a return-to-center behavior over time, so that if students do not keep sending messages, it decays back to a normal, middle setting.

While students initially found this feature intriguing, and they played with it a bit at the beginning of the semester, it did not prove that useful during the course.

5. Conclusions and Future Work

While we have found `WUTexter` to work well in our courses, we discuss in this section some simple changes that would make the application easier to use.

- The form of the messages appears simple, but on most phones the messages require changes between character sets (letters, numerals, special symbols). The course name contains numerals, the account must be prefaced with an at-sign, and the main text of the message is typically alphabetic.
- The secret key, while useful in preventing intrusion, is a burden for students to remember and type.

We are currently addressing the above problems by developing a client-side app for the iPhone and the Android platforms, along the following lines:

- The app would sense whether messages should be sent via data plan (Internet) or texting (cell tower) based on which resources are available and which method is preferred by the user.
- The app could present the 5 buttons of the `iClicker` as shown in Figure 1, a text area for asking questions, and a $+$ and $-$ button to indicate confusion or boredom.

- The app could make some commonly asked questions available from a menu, so that typing them would be unnecessary.

Acknowledgments

The authors thank Roger Chamberlain for his feedback concerning the use of this tool. The authors also thank the many students who have shared their reactions and experience using `WUTexter`. We thank the reviewers for their feedback.

References

[1] “iclicker web site,” http://www.iclicker.com.

[2] A. Czapracki and C. Burrows, “Bringing students and faculty together through mobile devices,” in *Proceedings of the 39th Annual ACM SIGUCCS Conference*, ser. SIGUCCS ’11. New York, NY, USA: ACM, 2011, pp. 215–218. [Online]. Available: http://doi.acm.org/10.1145/2070364.2070420

[3] A. T. Chamillard, “Using a student response system in cs1 and cs2,” in *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’11. New York, NY, USA: ACM, 2011, pp. 299–304. [Online]. Available: http://doi.acm.org/10.1145/1953163.1953253

[4] T. J. Hickey and W. T. Tarimo, “The affective tutor,” in *Proceedings of the 2014 meeting of the Consortium for Computing Sciences in Colleges, Northeastern Region*, 2014.

[5] T. H. Harrison, D. L. Levine, and D. C. Schmidt, “The design and performance of a real-time corba event service,” in *Proceedings of the 12th ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA ’97. New York, NY, USA: ACM, 1997, pp. 184–200. [Online]. Available: http://doi.acm.org/10.1145/263698.263734

[6] T. Murphy, K. Fletcher, and A. Haston, “Supporting clickers on campus and the faculty who use them,” in *Proceedings of the 38th Annual ACM SIGUCCS Fall Conference*, ser. SIGUCCS ’10. New York, NY, USA: ACM, 2010, pp. 79–84. [Online]. Available: http://doi.acm.org/10.1145/1878335.1878356

The Value of Video Quizzes in a Computer Science Flipped Classroom: An Empirical Study

Lisa L. Lacher
Department of Computer Science
University of Houston – Clear Lake
Houston, Texas, USA
llacher@uhcl.edu

Mark C. Lewis
Department of Computer Science
Trinity University
San Antonio, Texas, USA
mlewis@trinity.edu

***Abstract*—Flipping the classroom has become much more common in the last several years, but in the field of Computer Science it is still a much rarer means of delivery than the traditional classroom setting. One big concern is whether or not students will spend the preparation time necessary outside of the classroom in order to come prepared to class ready to engage in the active learning activities prepared by the instructor and thus maximize their learning potential. The study concerns four class sections in a Principles of Computer Science I course. Each student was required to watch video lectures prior to class. Some students were also required to take pre-class video quizzes which provided those students with immediate feedback on the level of their preparedness. Statistical analyses did not support the hypothesis that the pre-class video quizzes were effective in helping the students earn better grades. Student perceptions of the usefulness of video quizzes and anecdotal evidence from instructor experience are also presented.**

Keywords-education; computer science; flipped classroom, assessment

I. Introduction

The ability for faculty to easily record and post video content, combined with the capability for students to view that content on many platforms and in many locations has led to a significant increase in the use of flipped classrooms. In flipped or inverted classrooms, students learn new content from sources such as video lectures which prepare them to work on assigned problems in the classroom. The basic idea behind flipped classrooms is that more classroom time can then be dedicated to active learning where the instructor can provide immediate feedback. However, the success of the flipped classroom is highly dependent on students actually taking time outside of class to do the required preparation. In this paper we look at the impact of having regular quizzes over the video lectures compared to just having verbal admonitions from the instructor.

The flipped classroom approach has been used in a variety of disciplines, including computer science, although it appears that more instructors have experimented with this approach in upper division courses. Gehringer and Peddycord [1] used the flipped model in an upper level Computer Architecture course. Sureka, Gupta, Sarkar, and Chaudhary [2] as well as Gannod, Burge, and Helmick [3] experimented with a junior level Software Engineering course. Amresh, Carberry, and Femiani [4] did evaluate the effectiveness of the flipped classroom approach for teaching CS1. Their early observations on the success of this model were mixed. These studies essentially report on the initial success of the flipped approach in these classes in terms of engagement and grades. However, a major factor in this success or lack of success is how well the students prepare outside of the classroom so that they can take maximum advantage of the activities that take place during class time. Our work also takes place in CS1 and is taught using a flipped methodology. In particular, we are evaluating whether or not gate-check quizzes help students better prepare outside of the classroom -- to ensure that they had the background information and knowledge necessary to effectively engage and participate in the flipped class session's activity. This paper will report the preliminary findings of our study to understand if gate-check quizzes help beginning programming students be better prepared.

The remainder of this paper is organized as follows: Section II provides a general background of flipped classrooms and the course framework. Section III details the design of the experiment that was conducted to investigate the usefulness of video quizzes as a tool for motivating students to watch the videos. Section IV presents an analysis of the results from the experiment. Section V gives a summary and the conclusions.

II. Background

The use of lecture as the primary method of imparting information to students during class time has a long history. Teaching Computer Science courses has typically involved the instructor lecturing and conveying information while the students were responsible for listening and note-taking. Current research into how students learn challenges the traditional approach and suggests that students should be more actively engaged with the material in the course to help maximize their learning. The National Research Council has stated that "….the new science of learning is beginning to provide knowledge to improve significantly people's abilities to become active learners who seek to understand complex subject matter and are better prepared to transfer what they have learned to new problems and settings" [5].

There are many active learning strategies in literature; however, incorporating these active learning strategies in class means that the instructor has less time to convey the course content required for the learning activities. One pedagogical model that has emerged to address this issue is the flipped or inverted classroom.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

A. Flipped Classrooms

The fundamental idea behind flipping the classroom is that more classroom time should be dedicated to active learning where the teacher can provide immediate feedback and assistance. In this model, students are first exposed to the material they are studying outside of class using video lectures or by reading a textbook. During the class time, they usually do problem solving exercises or discuss the topics they were just exposed to. When we look at Bloom's revised taxonomy [6] (shown in Figure 1), we can see that the goal of this method is for the students to do the lower levels of cognitive work outside of class and then focus on the higher levels of cognitive work in class where they can get assistance from the instructor or their peers. This is quite different from the traditional classroom model where the students are first exposed to the material during class time via lectures and the higher levels of cognitive work are performed outside of class through homework.

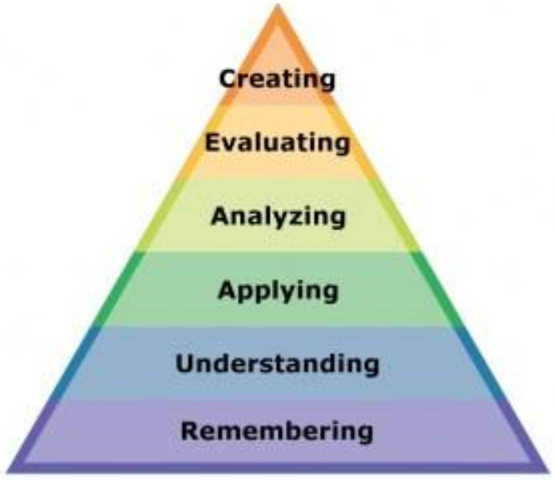


Figure 1 Bloom's Taxonomy (Revised)

Carnegie Mellon University's just-in-time lecture project suggests that video-based education supports the same level of learning effectiveness as face to face instruction [7]. Zhang et al. assessed the learning effectiveness of video on learning effectiveness and found that simply incorporating video is not always sufficient to improve learning [8]. They suggest that interactivity can be used to improve learning effectiveness. The student must take the initiative outside of class to prepare for class. Not all students, especially students in beginning level course have the self-discipline to take on this responsibility without some help from the instructor. Not all instructors have the technology available to them to create interactive videos. Will having a pre-class quiz increase the likelihood that students will use out-of-class time to watch the videos in order to learn the material necessary to be successful? Educators have suggested that they will help assure that students are prepared and will earn better grades, but provide no proof of the success of this practice [9, 10].

B. Course Design

The CS1 course, CSCI1320 – Principles of Computer Science, includes an introduction to the Scala programming language, Linux command line and covers sequence, selection, branching, iteration, higher order functions, and collections in the first half of the course. This course is required by students majoring in computer science, engineering, physics, and mathematics. It is an option in the Biology and Geoscience majors and counts as a possible course in Trinity's Common Curriculum so that many students consider taking it for reasons outside their major. The four sections of this course are designed to be an initial introduction into learning a programming environment. Three of the sections met two times per week for seventy-five minutes. Our initial study only concerns half of a fifteen week spring semester for a total of fifteen meetings. The last section met three times per week for fifty minutes and was held in the previous fall semester. This study is also only concerned with the progress of the class during the first half of that semester.

Author Lewis began flipping his sections of CS1 and CS2 in the fall of 2012 using video lectures posted on YouTube. Every chapter of the textbook had several videos associated with it: ranging from four to twelve videos per chapter. Students were required to watch a total of 53 videos (13 hours and 52 minutes total time) before the midterm. Videos range from seven to twenty-six minutes in length. All in-class activities were problem-based exercises that required the students to write code, based on certain pre-defined sections of the textbook and videos per day. Students were free to work alone or collaborate on all in class activities. Solutions to all exercises were discussed by the instructor during the class and all code written in class was posted on the course website so that students could study it further after class. At the end of the class period, students were required to fill out a minute essay, part of which included giving feedback to the instructor on subject matter that they wished would be further explained in the next class period.

Our study is motivated by our desire to improve the course and our hunch and student comments about the amount of time students are actually spending on watching the videos and preparing outside of class. We hypothesize that this shortcoming could be addressed with gate-check quizzes.

III. Study Design

The study is designed to answer the question of whether the use of "gate-check" pre-class video quizzes is helpful in improving the students' learning as measured by the in-class quiz and midterm exam scores. The study basically compares the performance of students that were required to complete pre-class video quizzes that provided immediate feedback about their level of preparedness with the students that did not use any sort of pre-class quiz to self-evaluate their preparedness for the next active learning activities.

The experiment utilizes a Non Equivalent Pre-test Post-test Control group design consisting of a control group (no pre-class video quizzes) and an experiment group (required pre-class video quizzes) to help to prepare for the next active learning activities. The details of the study are given below.

A. Hypotheses

The following hypothesis was posed in the study

Hypothesis 1: The students using the gate-check video quizzes produced higher in-class quiz and midterm exam grades than their peers who did not have to take the gate-check video quizzes.

Hypothesis 2: The students with the lower aptitude test scores would benefit more (earn higher grades) using the gate-check video than their lower aptitude peers who did not have to take the gate-check video quizzes.

1) Independent and Dependent Variables

The experiment manipulated the following independent variable:

- ***The pre-class video quiz technique.*** Subjects either took a pre-class quiz over the required video content for the course or did not take any pre-class quiz.
- ***The computer aptitude test.*** Subjects took a computer aptitude test during the first week of the courses.

The following dependent variables were measured:

- ***Student In-class Quiz and Midterm exam performance.*** These measures include the student grades on the three in-class quizzes and a midterm exam.

B. Participating Subjects

Sixty-three undergraduate students enrolled in the Principles of Computer Science I course at Trinity University participated in this study. These students were in four different class sections taught by two different instructors (each taught two sections). Each section had between fifteen and eighteen students enrolled. One section was in the fall of 2013, and the other three were in the spring of 2014. The course was taught in a flipped classroom style and required that the students watch video lectures and read required textbook sections before attending class.

C. Artifacts

Each of the students had taken three quizzes and a midterm exam after seven and one half weeks of class. All sections used identical written tests and quizzes which were graded by a student grader using a rubric developed by the instructors for consistency, except for the Fall 2013 course which used different in-class quizzes that are not included in this study (only midterm grades were compared). Although one class was during the previous semester, the same schedule (as seen in Table I) was followed. During this time the students were required to watch the videos and read the text associated with each day's in-class active learning activities. Every day in class the students worked on writing code solutions for a variety of problems posed by the instructor. On some days the class

TABLE II. COURSE SCHEDULE

Date	Topic	Reading	Due
1-16	Introduction to Class and the Future of Computing	Ch. 1	
1-21	Future of Computing and the Linux Command Line, Linux, vi and Scala	Ch. 2.1-2.3	
1-23	More Command Line, Linux, vi and Scala	Ch. 3.1-3.3	
1-28	Show Your Code with Scala Expressions and Types; More Command Line, Types and Methods; Binary Arithmetic	Ch. 3.1-3.5	IcP #1 (Chapter #2 Exercises)
1-30	Strings and Variables, Sequential Execution and Scripts	Ch. 3.6-3.9	Quiz #1
2-4	Boolean Expressions and if	Ch. 4.1-4.3	
2-6	Boolean Expressions, Functions, and Function Literals	Ch. 4.4-4.5, 5.1-5.2	
2-11	Show Your Code with Higher Order Functions	Ch. 5.3-5.5	IcP #2 (Chapter 4 Projects)
2-13	Higher Order Functions and Recursion for Repetition	Ch. 5.6-5.7 Ch. 6.1-6.4	Quiz #2
2-18	Show Your Code and Recursion, Match and Patterns	Ch. 6.5-6.7	IcP #3 (Chapter 5 Projects)
2-20	Collection Types (Arrays and Lists)	Ch. 7.1-7.3	
2-25	Collection Methods	Ch. 7.4-7.6	Quiz #3
2-27	Argument Passing and While Loops	Ch. 7.7-7.11 and Ch. 8.1-8.2	Assignment #1 (Chapter 6 Projects)
3-4	Show Your Code with Loops	Ch. 8.3-8.6	IcP #4 (Chapter #7 Projects)
3-6	Midterm		

started off with an Interclass problem (IcP). Here students were required to code solutions to textbook project selections of their choice prior to coming to class and then show their solutions to the class. On other days, the class started with an in-class quiz which was graded and returned to the students the following class period. There was one assignment during the first half of the semester and this was worked on by the students outside of class.

D.Experimental Procedure

To evaluate the hypotheses posed in Section III. A, the study contained a control group and an experiment group. Figure 2 provides an overview of the procedure followed.

The details of the experiment steps are given below.

1) Step 1 (Course Assessments with and without Video Quizzes): The grade for the video quizzes needed to be included in the overall grading for the course. Before the video quizzes were introduced, course grading was calculated as follows: four assignments were worth forty percent of the grade, two tests (midterm and final) were worth thirty percent of the grade, five quizzes were worth ten percent of the grade, Interclass problems were worth ten percent of the grade, and class participation was worth ten percent of the grade. With the introduction of the video quizzes, it was decided to reduce the assignment and test percentages, thus in the courses with the video quizzes the grading was calculated as follows: four assignments were worth thirty five percent of the grade, two tests (midterm and final) were worth twenty five percent of the grade, five quizzes were worth ten percent of the grade, Interclass problems were worth ten percent of the grade, twenty six video quizzes were worth ten percent of the grade, and class participation was worth ten percent of the grade.

2) Step 2 (Pre-Test): A twenty-six item pre-study computer aptitude test, an eighteen question learning survey, and a nineteen question general information survey were administered at the beginning of the semester. The Fall 2013 course did not take these tests so the students in that course were not considered for that part of the study.

a) Computer Programming Aptitude Test: This is a hybrid test created by the University of Kent[11] that is comprised of elements involving numerical problem solving, logical reasoning, attention to detail, pattern recognition, and the ability to follow complex procedures. This test is very appropriate for Computer Science I students because this test does not require any knowledge of programming. Numerical problem solving is similar to the logical thinking and trouble shooting required in programming. Pattern recognition is necessary in understanding the graphical representations of symbols and procedures and leads to the attention to detail required to do things such as find misspelled variable names or missing semi-colons. The ability to follow complex procedures is necessary to follow coded instructions, sequences, branching, and repetitions of events in order to decide how one set of instructions affects another and the flow of events. The students first took an untimed, easy three question practice test before they took the timed (forty minutes) twenty-six question test.

b) Learning Approach Survey. This eighteen item survey was developed by Dolmans, Wofhagen, and Ginns [12] and provides a valid and reliable way to measure students' learning approaches (deep vs. surface approaches). They state that "based on intrinsic interest in the topic, students taking a deep approach try to understand ideas and seek meaning and understanding. Often driven by a fear of failure, students adopting a surface approach are focused on meeting external requirements, such as assessments, particularly through rote learning. A deep approach is assumed to positively correlate with achievement and a surface approach is assumed to negatively correlate with achievement." Because the objective of this study was mainly to investigate the effect of video quizzes on learning effectiveness, learner characteristics were not considered as independent variables for this study. We collected this information for future work.

c) General Information Survey. In this survey students were asked questions such as their gender, previous programming experience, previous computer experience, confidence in their technical skills, comfort with this computer science course, grade expectations, time they spend on extracurricular activities, work, and caring for others, how they rate their time management skills, whether they like to work collaboratively or alone, whether or not they were ever encouraged to take computer science courses, and their nationality. This information was collected for future research.

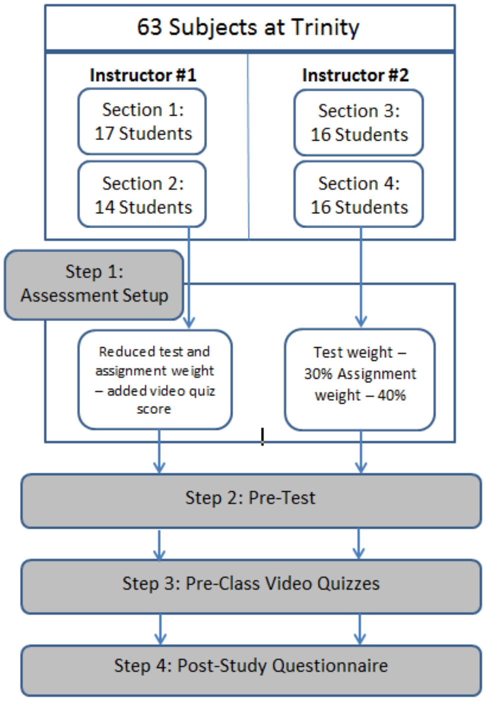


Figure 2 Experiment Design

3) Pre-Class Video Quizzes: Two sections of students were required to complete a quiz based on videos that covered certain sections of the text (see Table I) prior to the start of class. The number of questions on each quiz ranged from four questions on the shortest quiz, to eleven questions on the longest quiz. There were generally one to two questions asked per video. The quiz questions were one of the following: true/false, multiple choice, or matching. The video quizzes were given via our learning management system which allowed them to be auto-graded and closed at class time each day. The quizzes themselves typically focused on things like syntax that would have been difficult for students who did not watch the videos or read the book, but their answers were straightforward after watching the videos. The questions were also presented in the order that material appeared in the videos. The students were allowed three attempts on the first six video quizzes, and then based on class feedback, it was modified to two attempts per quiz for the last six quizzes.

4) Post-Study Questionnaire*:* A fifteen question post-study questionnaire was administered to the students after the midterm. The post-study questionnaire collected feedback from the student regarding their usage of the videos and the text, and their perception of the effectiveness of video quizzes or lack of video quizzes.

IV. Analysis and Results

Statistical analyses were carried out on the data collected during the study. The hypotheses described in Section III A were tested to determine the usefulness of the video quizzes in improving student understanding as proven out as grades. An alpha value of 0.05 was selected for judging the significance of the results. The post-study questionnaire was also used to gain an understanding of the student's perception of the usefulness of the video quizzes to their learning.

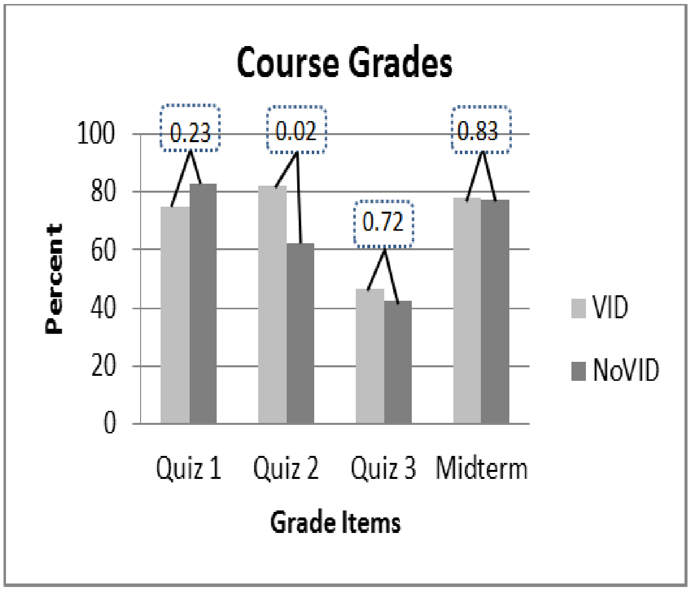


Figure 3. Comparing Performances of students who took the video quizzes (VID) and student who did no use the video quizzes (NoVID) on In-class quizzes and Midterm

A. *Hypothesis 1: The comparison of students using the gate-check video quizzes to produce higher in-class quiz and midterm exam grades to their peers who did not have to take the gate-check video quizzes.*

This section provides analyzes of the effect that video quizzes had on the student in-class quiz and midterm exam grades. The marks scored by the students that used video quizzes and the marks scored by the students that did not use video quizzes were compared. Each of the scores was tested using an independent samples t-test.

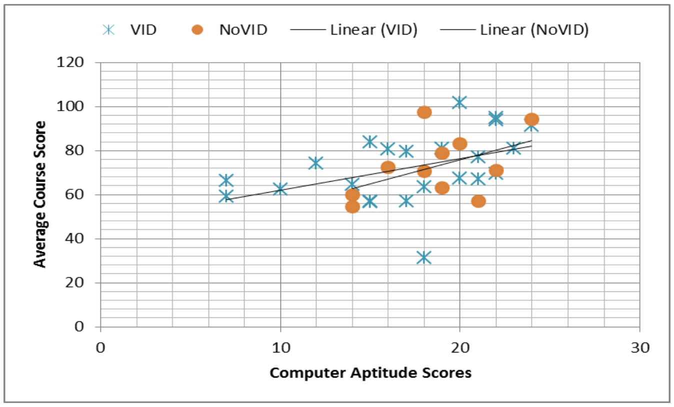


Figure 4. Comparing Total Scores of the students who took video quizzes (VID) and the students who did not take video quizzes (NoVID)

Figure 3 shows comparisons of the average scores of students who took the video quizzes (VID) and students who did not use video quizzes (NoVID) on Quiz1, Quiz2, Quiz3, and Midterm. On Quiz1 the NoVID performed better than the VID, while the VID performed better than the NoVID on the others. On the Midterm, the VID did slightly better than the NoVID. However; the p value from the t-tests are illustrated in Figure 3 and do not show a significant difference in the average scores of VID and NoVID students for three of the four grade items: Quiz 1 p= 0.23, Quiz 3 p=0.72 and the Midterm-p = 0.83 are all greater than the alpha value of 0.05.

In comparing the mean over the quiz and midterm deliverables using a simple t-test, the NoVID mean of 73 was slightly better than the VID average score of 72.2. However; given a NoVID standard deviation of 15.82 and VID standard deviation of 14.36 and using an alpha=0.05, the difference between the average grades for the two groups (VID vs. NoVID) was not significant.

An independent samples t-test reveals that the NoVid student teams performed as well as the VID student teams on grades. These results indicate that using the video quizzes is not beneficial to grade performance.

B. *Hypothesis 2: The students with the lower aptitude test scores would benefit more (earn higher grades) using the gate-check video than their lower aptitude peers who did not have to take the gate-check video quizzes*

This section provides analyzes of the effect that video quizzes had on the grade performance of students with different computer aptitudes. Figure 4 shows each student's aptitude and compares the scores of the students who took the video quizzes with those who did not. It can be seen that regardless of whether or not the students took the video quizzes, increased aptitude seems to correlate with increased course performance. Our next question is then if the video quizzes had any possible effect?

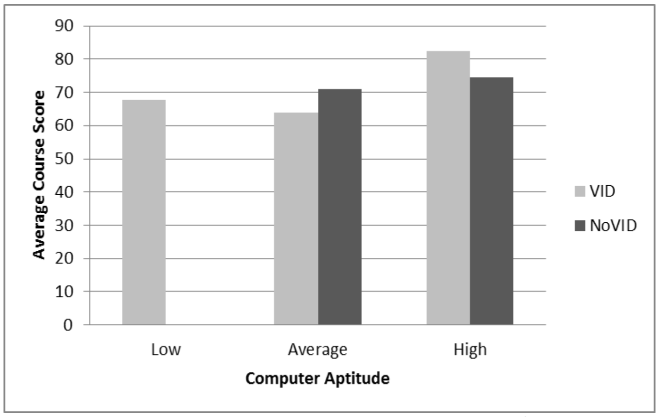


Figure 5 Student Computer Aptitudes vs. Grade Performance

The University of Kent's Computer Programming Aptitude Test scores are divided into three categories. Nineteen or above is considered an above average test score. People with these test scores are most likely ready for the challenge of computer careers. Scores between thirteen and eighteen are considered average. Even though the test is only a rough guide, people who scored within this range may find it more difficult to do well as a computer programmer. Scores less than thirteen are considered below average.

To help analyze this hypothesis, we divided the students into the three categories: high = students with scores nineteen or above, average = students with scores between thirteen and eighteen, and low = students who scored twelve or less. This formed three groups for each of the VID and NoVID categories. It is interesting to note that the average computer programming aptitude score in the above average category (High) for the both the students who took the video quizzes (VID) and the students who did not take the video quizzes (NoVID) was 21. It is also interesting to note that the average computer programming aptitude score in the average category for both the students who did take the video quizzes (VID) and those that did not (NoVID) was 16. There were no students who did not take the video quizzes (NoVID) in the Low category. The average computer programming aptitude score for the students who scored below average (Low) and took the video quizzes was 9.

Figure 5 shows that the VID students performed better than the NoVID students and after performing a t-test it was shown that the grade difference was statistically significant (p = 0.025), thus the video quizzes seemed have an effect on grade performance. Although both the VID student and NoVID students in the average aptitude category had exactly the same average aptitude, the NoVID students performed better than the VID students although the grade difference was not statistically significant. However, it appears that the video quizzes did not help students with average computer programming aptitude. Although there were no students in the NoVID group that had below average computer programming aptitudes, it is interesting to note that the lower aptitude VID students performed better on grades than the average aptitude VID students, although they did not perform better than the average aptitude NoVID students. It is possible that the video quizzes may have helped these below aptitude students because their computer aptitude scores suggest they these students should find it much more difficult to perform well in this class, yet they performed close to the average NoVID students and surpassed the average VID students. We attempted to tease out possible answers to this type of question through our student post study survey.

C. *Student perceptions*

The usefulness of the pre-class video quizzes used by students was also evaluated using feedback from the subjects in the both the experiment group (the members of VID) and the control group (the members of NoVID). The quiz questions fell into three categories: how the students used the quizzes, how the students used the textbook, and their perception on whether or not the quizzes were useful. Because this study primarily is concerned with the video quizzes, we will look primarily at the eight questions that dealt with those. It is apparent from the question results shown in Table II that the video quizzes did cause the students to watch the videos more often and watch them more thoroughly—which was one of the fundamental goals of having the students take the video quizzes.

TABLE II. VIDEO USAGE

Did you watch the video lectures?		
	VID	**NoVID**
Rarely	15%	8%
Sometimes	20%	58%
Often	25%	17%
Almost Always	40%	17%
Did you review portions of the lecture that seemed unclear?		
Rarely	15%	33%
Sometimes	35%	50%
Often	30%	17%
Almost Always	20%	0%
Percentage of videos generally watched for each class meeting:		
None	5%	0%
25%	10%	58%
50%	20%	25%
75%	40%	17%
100%	25%	0%

It can also be seen in Table III that both the students who were required to take the video quizzes (VID) and the students who were not required to take the video quizzes (NoVID) felt that they were beneficial to their learning and did (or would have) caused them to spend more time preparing by watching the videos or read the text. However, the last question on the post survey asked the students about their general thoughts on the video quizzes and a majority of the students did not like the video quizzes. Some stated that they did not feel that the quizzes really prepared them for the class, that the quizzes took too long to take, that time could have been better spent than watching the videos, and that the video quizzes did not actually help them remember any information. One of the post survey questions asked the students whether they thought it would be more helpful to have a small coding assignment to practice the video content rather than the video quizzes: sixty percent of the VID students and sixty seven percent of the NoVID students would prefer a small assignment. The implications and the discussion of these results are presented in Section V.

TABLE III. VIDEO QUIZ MOTIVATION

The pre-class quizzes were beneficial to your learning in the course		
	VID	**NoVID**
True	70%	NA
False	30%	NA
I would have watched the videos if we were not (would have) been required to take quizzes on the material.		
True	55%	58%
False	45%	42%
Having quizzes caused me (would cause me) to watch the videos or read the book more than I would have otherwise.		
True	75%	75%
False	25%	25%

V. Summary And Conclusions

In flipped classrooms, students learn course content from sources such as video lecture which is intended to prepare them to work on assigned problems in the classroom. A major success factor, when using the flipped approach, hinges on how well the students come prepared to class. Our primary hypothesis is that pre-class video quizzes would help students earn better grades. We assumed that earning better grades would correlate with being better prepared. However, our experimental study does not support this hypothesis. We speculate that the quizzes failed to elevate grades because they did not facilitate deep learning. Students focused on answering the multiple choice type questions and did not delve deeper into the material. While one of the instructors felt that the quizzes did improve students preparedness for class, this was not seen in both sections that were required to take the quizzes and could be due to many other factors unrelated to the quizzes.

Our second hypothesis concerns assertions that having a mechanism to help ensure that students watch the videos would help students with lower computer programming aptitudes perform better in the course. Results indicate that to the contrary, it was the higher aptitude students who benefited the most from having the video quizzes. We speculate that higher aptitude students are more prone to slacking off on tasks like watching the videos, so a simple tool that enforces them spending the time helps them more. Unfortunately, raising the performance of the high aptitude students, while commendable, is something that we view as less significant than bringing up the lower end students. It is possible that the tendency for this style of quiz to lead to more surface learning was detremental to the lower aptitude students and that another style of enforcing preparation that aims to facilitate deep learning would be superior.

We plan to test the use of small programming tasks, instead of multiple choice quizzes, in the future to see if that is a superior approach. We believe that style of preparation enforcement would lead to deeper learning and benefit the middle and lower aptitude students more.

Acknowledgment

We thank the students in the Principles of Computer Science I course at Trinity University for their participation in this experiment.

References

[1] E.F. Gehringer, and B.W. Peddycord III, “The inverted-lecture model: a case study in computer architecture”. Proceeding of the 44th ACM technical symposium on Computer science education. ACM, 2013.

[2] A. Sureka, M. Gupta, D. Sarkar, V. Chaudhary, “A Case-Study on Teaching Undergraduate-Level Software Engineering Course Using Inverted-Classroom, Large-Group, Real-Client and Studio-Based Instruction Model” CoRR abs/1309.0714, Sep 2013

[3] G.C. Gannod, J.E. Burge, and M.T. Helmick, “Using the inverted classroom to teach software engineering.” Proceedings of the 30th international conference on Software engineering. ACM, 2008.

[4] A. Amresh, A.R. Carberry, and J. Femiani, “Evaluating the Effectiveness of Flipped Classrooms for Teaching CS1”. Proceedings of the Frontiers in Education Conference, IEEE, 2013

[5] National Research Council (2000). How people Learn: Brain, Mind, Experience, and School. Washington, D.C.: National Academy Press, p13

[6] H. Coffey. (2008). *Bloom’s Taxonomy* [Online]. Available:http://www.learnnc.org/lp/pages/4719

[7] Carnegie Mellon http://www.jitl.cs.cmu.edu

[8] D. Shang, L. Zhou, R. O. Briggs, J.F. Nunamaker Jr., “Instructional video in e-learning: Assessing the impact of interactive video on learning effectivenss”. Information and Management 43, (2006) 15-27

[9] S. Zappe, R. Leicht, J. Messner, T. Litzinger, and H.W. Lee, ““Flipping” the Classroom to Explore Active Learning in a Large Undergraduate Course”, American Society for Engineering Educagion, 2009

[10] R. Toto and H. Nguyen, “Flipping the Work Design in an Industrial Engineering Course”. ASEE/IEEE Frontiers in Education Conference, Oct 18-21, 2009, San Antonio, TX.

[11] University of Kent (accessed 04/18/2014). *Computer Programming Aptitude Test* [Online]. Available: https://www.kent.ac.uk/careers/tests/computer-test.htm

[12] Dolmans, D., Wolfhagen, I., & Ginns, P. (2010). Measuring approaches to learning in a problem-based learning context. International Journal of Medical Education, 1, 55-60

Tracking Attendance Pilot

D. Deugo[1]
[1]The School of Computer Science, Carleton University, Ottawa, Ontario, Canada

Abstract – *Students experiencing difficulty with their studies often don't identify themselves as students in need of support. The Science Student Success Centre at Carleton University actively seeks out these science students to give them the help they require. One approach is to look at a student's grades as an indicator of need. Another indicator, along side of grades, is attendance in class. In this paper we look at the usage results of piloting an attendance tracking system in two classes, both containing less than 50 students. While our online card swiping attendance tracking system is usable for small class sizes, we also discuss the groundwork of our attendance tracking Android mobile application that is suitable for much larger classes.*

Keywords: Attendance, Tracking, Android, Pilot

1 Introduction

In [1], we described the initial view of our attendance tracking system. Our motivation for building the system came from The Science Student Success Centre (SSSC) at Carleton University. The SSSC collaborates with other on-campus departments to offer workshops, events, and activities that help to develop a student's academic and professional skills. The SSSC also takes an active role in helping science students struggling in their first-year computer science, math, and science courses with the mission as follows:

- Increase the engagement and retention of students in the Faculty of Science
- Foster the growth and achievements of high performing students
- Identify and support students who may be experiencing difficulty with their studies
- Inform students of professional and academic development opportunities
- Aid in the recruitment of outstanding students to the Faculty of Science

One of the questions for the SSSC was how to determine students in need of its services. One indicator it used was to look at first term, mid-term marks in late October. We also believe that looking at students' attendance records for the same period provides another key indicator for students in need. The SSSC already contacts students with marks less than 60% and asks them to come in and visit with the SSSC team to talk about the issues they are facing with their courses and possible techniques and actions they can take to overcome the issues. We also felt that looking for students with less than 60% attendance records was another group we should meet with individually and that there would likely be a crossover between low grades and low attendance students.

Before the SSSC could track low attendance students, we needed to build an attendance tracking system and pilot its usage in classes to examine its performance and usability. In this paper we look at the results of piloting the attendance tracking system in two classes, both containing less than 50 students. Taking attendance using paper and pen was one approach we could have used for these classes, but we knew it was slow and prone to errors. In addition, the paper method required a data entry phase in order to generate reports, which also suffered from the same problems. Therefore, using pen and paper was ruled out from the beginning.

1.1 Goals

Reiterating our goals and objectives from [1]. Our main goal was to provide a fast and efficient attendance tracking system. In addition, the system must work in any and all classrooms at Carleton University, including its electronic classrooms – those with computers and projectors – and those classrooms containing no computers. Finally, a further goal was to provide a system that requires minimum hardware, and can be maintained at minimum cost.

1.2 Objectives

To meet the goals, we had the following objectives:

- Use easily found, inexpensive hardware
- Make use of students' mobile devices to help with the attendance tracking process
- Use open source software to minimize development and maintenance costs
- Installation, either for end users or system administrators, should be fast and simple
- Attendance reports should provide end users with the greatest flexibility for manipulating the collected attendance data
- Support both Mac and PC platforms, and Android and iOS mobile devices

1.3 Outline

In section 2 we further describe literature on existing attendance tracking systems. Section 3 describes our current

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

system and how it was used in the classes to take attendance in our pilot study. Section 4 describes the results of our pilot. Section 5 provides our conclusion and identifies our future work.

2 Background

Attendance tracking has been used in universities and colleges as a means of improving attendance, retention rates, and grades [1]. As [2] points out in their conclusion:

> *"The results of this study indicate that attendance does matter for academic achievement in a Principles of Economics courses. The evidence suggests that the effect is nonlinear, becoming important only after a student has missed four classes during the semester. That really seems to mater is excessive absenteeism."*

Attendance tracking is also used for other purposes and in different domains. For example, in [8] a system that deals with attendance tracking in remote locations is described.

> *" Hyke combines voice-biometrics with accurate location tagging for tracking attendance in remote locations without the need for a trusted mediator on-site. Hyke was designed based on our observation of a currently deployed teacher attendance tracking system in rural Rajasthan, India. We have implemented some of the key components in Hyke, and discuss some of the security concerns in the system. The Hyke biometric stack for voice recognition is built atop several open source technologies, and provides a simple interface for non-expert users. Our evaluations with Indian speakers over telephone audio suggests the biometric stack is at par with the current state of the art. We believe this will be a useful tool for researchers who would like to incorporate voice technologies in their developing world projects"*

In [1] we reported on several attendance tracking systems specifically built for such a purpose. However, one can also use generic systems, such as Google Docs and its Google Forms capability, and build a simple tracking and reporting system [3,4]

It is interesting to see that organizations, such as Heriot Watt University, go so far as to providing a policy on student attendance. In [5], they outlined four aims and objectives to developing their policy:

> *"2.1.1 Support and encourage students to achieve their full potential in their studies and in developing lifelong skills and competencies;*
>
> *2.1.2 Enhance the overall learning environment by encouraging active participation through regular attendance in all areas of study;*
>
> *2.1.3 Provide guidance on why attendance is important;*
>
> *2.1.4 Use attendance as one of the factors to be taken into account in reviewing a student's commitment to, and performance in, their academic studies and in developing an appropriate response to specific concerns about performance."*

Glassgow Caledonian University is another institution with such a policy, approved in June 2013. In their policy, they outline several principles outlining both the student and university's commitment to attendance:

> *"a. Students are active participants in their learning experiences and must take responsibility for achieving their potential through successful completion of each stage of their studies.*
>
> *b. Monitoring attendance can provide an indication of student commitment, motivation and any difficulties which need to be addressed.*
>
> *c. Regular attendance and academic achievement are closely linked. Students who attend regular student-supervisor meetings and researcher personal, professional and research training and development are more likely to enjoy a rewarding experience in which core skills and abilities, such as team-working, are developed.*
>
> *d. Schools will ensure that effective mechanisms are in place to both identify students considered to be at risk and offer appropriate support and guidance."*

Are these policies important? We believe the answer is yes as do others. As reported in [10] and outlined in [6]:

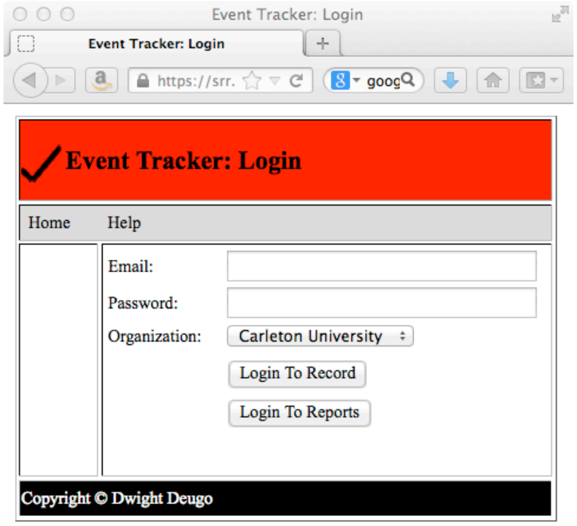
> *"The impact of attendance policy was significant. When attendance was explicitly required, 80% of the students missed 4 times or less and less than 1% missed 8 times or more. When the attendance policy was non-required and implicit, 73% of the students missed 4 times or less and almost 7% missed 8 times or more. When the attendance policy was explicitly non-required, only 52% of the students missed 4 or fewer times while 18% missed 8 times or more. The conclusion seems to be that professors do*

> *get the type of attendance that they encourage by the policy that they adopt."*

Is tracking and increasing attendance enough to help increase a student's academic performance? In the study reported in [7] attendance is "clearly associated with academic performance". However, like [9], it is also believed that student engagement "to the extent in which the student is involved with his/her studies" also plays and important role.

3 Approach

There are two types of people that interact with our attendance tracking system. The first type of person is called a user. A user is a person that is involved with taking attendance and reviewing the results. To login to the system, a user is presented with the screen shown in Figure 1.



Email:

Password:

Organization: Carleton University

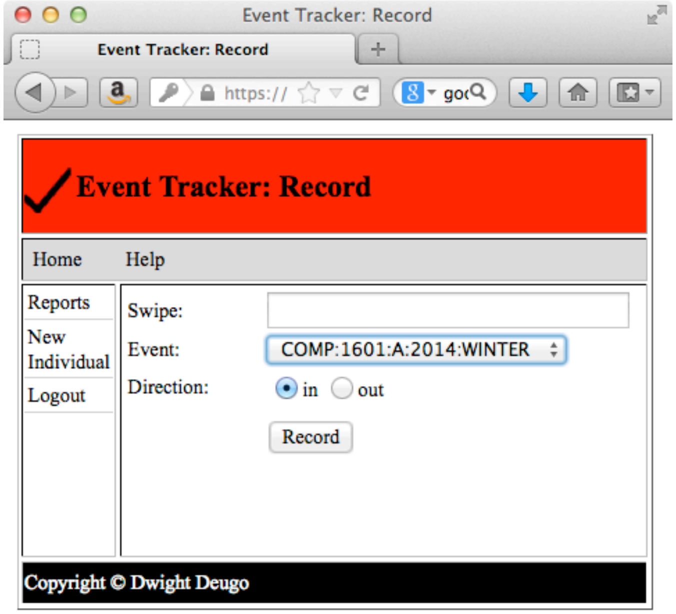
Figure 1: Login Screen

After entering their email address and password, a user can choose to login to either recording attendance or look at their attendance records for their associated events. A user must also identify the organization they are with, such as Carleton University. The system's administrator assigns a user's email address, password, organization, and the events they are tracking. The system administrator has access to the backend of the system and is outside the scope of this paper.

After providing the correct credentials and selecting Login To Record, a user is presented the screen shown in Figure 2. In this screen the user selects the corresponding event they wish to record attendance for, enters an individual's swipe information and then selects the Record button to record that individual's attendance in the event for the current time.

An individual is a person that is having their attendance tracked. They have a first and last name, an email address, and id and swipe information. Swipe information is data that can be read from an individual's id card, such as a student card, and is often, but not always, different from their student id or number.

The attendance tracking system has been built to accommodate two different types of organizations. The first is an organization that allows swipe information to be validated against a backend system and information returned from it about the corresponding individual to the attendance tracking system. The second type of organization is one where individual information based on swipe information is not available. In the latter case, every individual that will be tracked by the system must have his or her information entered manually into the system by a user. A user does this by selecting Login To Records from the login screen or simply selecting Record in the left hand menu from any screen once they are logged in. In fact, if a user attempts to take the attendance using swipe information that can not be associated with a given individual, they are taking directly to the Individual Registration Screen, as shown in Figure 3, automatically



Swipe:

Event: COMP:1601:A:2014:WINTER

Direction: ● in ○ out

Figure 2: Record Screen

Swipe information is entered in one of two ways. The first is manually typing in the information. This approach can be error prone, as there is significant amount of information on a standard identification card that is not in an easily human readable format [11] and it can be difficult to identify the information without a card reader. The second method of entering the swipe information is to connect a card reader to computer that is tracking attendance. Provided the card reader has a HID or Keyboard Emulation mode, once simply places the cursor in the swipe field, swipes the card in the reader and the information is automatically entered in the swipe field. The source of error in this situation is where the action of swiping the card was incomplete. Depending on the card reader, errors in swiping will produce an entry that can be recognized as an error. For example, the MagTek SureSwipe

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Reader, Model#: 21040145 produces an ;E? when a card is misread.

By selecting Reports in the left hand menu or selecting Login to Reports in the home screen a user is taken to the Reports Screen shown in Figure 4. All of a user's events are listed in this screen and by selecting one and clicking the download button, an Excel spreadsheet of results is downloaded for the user's review, as shown in Figure 5.

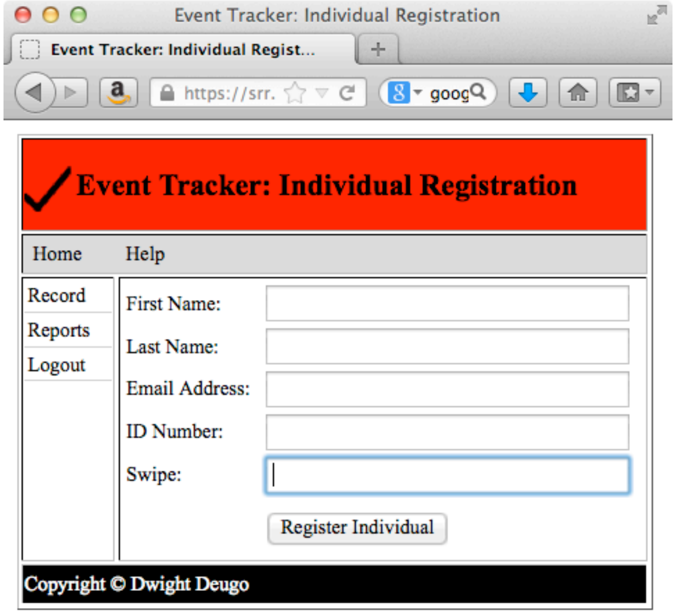
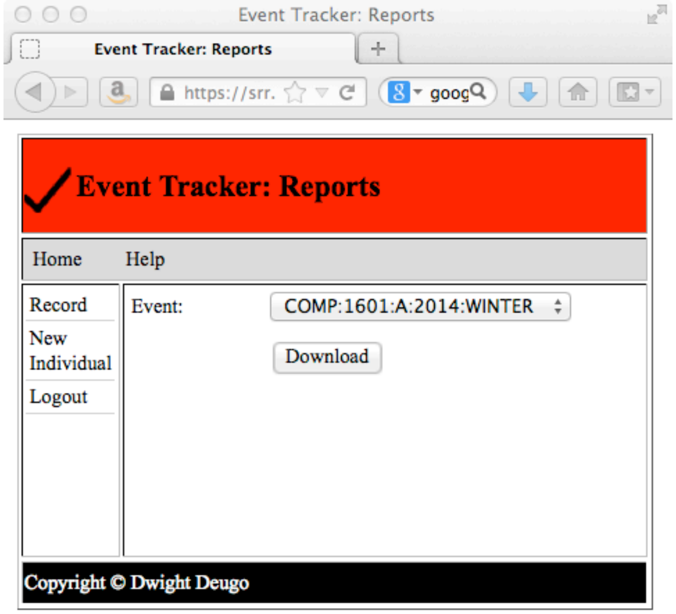
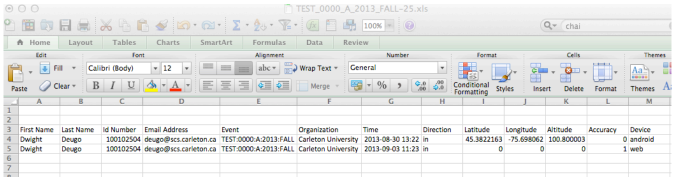


Figure 3: Individual Registration Screen



Event: COMP:1601:A:2014:WINTER

Figure 4: Reports Screen



	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3	First Name	Last Name	Id Number	Email Address	Event	Organization	Time	Direction	Latitude	Longitude	Altitude	Accuracy	Device
4	Dwight	Deugo	100102504	deugo@scs.carleton.ca	TEST:0000:A:2013:FALL	Carleton University	2013-08-30 13:22	in	45.3822163	-75.698062	100.800003	0	android
5	Dwight	Deugo	100102504	deugo@scs.carleton.ca	TEST:0000:A:2013:FALL	Carleton University	2013-09-03 11:23	in	0	0	0	1	web
6													
7													

Figure 5: Excel Report

Each entry in the Excel spreadsheet provides information on an individual checking in or out of an event. This information includes the following:

- First Name
- Last Name
- ID Number
- Email Address
- Event Name
- Organization
- Time
- Direction (In/Out)
- Latitude
- Longitude
- Altitude
- Accuracy
- Device

The Device entry can be Android, iOS or web, identifying the three methods of data entry available for the system. This paper discusses the web approach and the interface of our Android mobile device app. The only significant difference with the apps is that the GPS coordinates are taken off the mobile device and an accuracy percentage between 0 and 1 is calculated indicating where the device was used to identify an individual's attendance at an event. In the case of web entry, the accuracy is always 1 and the latitude, longitude and altitude are 0.

4 Results

Of significance to this paper is how quickly individual registration takes and how quickly individual attendance takes.

To answer the first question, we examine the time required to register and swipe in two different classes at Carleton University under two different scenarios. The first class had 41 students stand in line and register one after another. The instructor (user) performed all individual data entry. The total time required to register all 41 students was 2064 seconds for an average of 50.34 seconds per student. The second class had 31 students remain in their seats and come up one a time to register. The instructor performed all individual data entry. The total time required to register all 31 students was 3338 seconds for an average of 107.68 seconds per student. As mentioned, registration is a one-time operation. After a student (individual) is in the system, they can be swiped in for any and all events.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

To answer the second question we examine three different event-tracking scenarios. The first scenario was to swipe people in for the period starting 5 minutes before a class until 5 minutes after class started. In one class, we had 26 people arriving in this period for a total of 528 seconds processing time, taking an average of 20.31 seconds to process each student.

In another scenario, students stood in line at the end of class and were swiped out. In one class, we had 21 waiting for a total of 208 seconds processing time, taking an average of 9.9 seconds to process each student.

In the final scenario, student cards were collected in batches and swiped out during class. A typical outcome for processing 5 student cards required a total of 19 seconds processing time, taking an average of 3.8 seconds to process each student card.

Table 1 summarizes these results.

Table 1: Registration and Swipe Timings

	Reg. in Line	Reg. in Seats	Swipe Before Event	Swipe End of Event	Batch Swipe
Students	41	31	26	21	5
Total Processing Time (seconds)	2064	3338	528	208	19
Average Processing Time (seconds)	50.34	197.68	20.31	9.90	3.8

5 Conclusions

We believe our current web-based system can be used for tracking attendance in events of up to 50 people. At this number, a little less than 1 hour (50.34 seconds * 50 = 41.95 minutes) is required to register them with the system. This is a one-time operation that can be done before the start of the first occasion of an event or class. Moreover, if people attend different events, they do not need register with subsequent events as they are known to the system. The registration operation can be eliminated completely if the event organization provides access to an individual's information. The system has been developed with a plugin style architecture, where validation and retrieving of individual data can be done through accessing a local database or assigned a different software component that accesses an organization's remote database.

Swiping people's attendance at the start or end of an event is also within reasonable time limits. Swiping in 50 people before the start of an event takes (20.31 seconds * 50 = 16.93 minutes) based on our observed arrival times of people. Moving this process to the end of the event and swiping people leaving the event speeds up the process. In this case, the total processing time is decreased to less than 10 minutes (9.9 seconds * 50 = 8.25 minutes).

The web-based system breaks down as event sizes increase beyond 50 people. In tables 2 and 3, the timings for registration and swiping people at the start and the end of an event are shown. In cases of 100 and 200 people, the numbers show that processing times are unmanageable, except for the 100 people case of swiping at the end of an event.

Table 2: Registration Timings for Typical Event Sizes

Students	Average Processing Time per person (seconds)	Total Processing Time (seconds)	Total Processing Time (minutes)
50	50.34	2517	41.95
100	50.34	5034	83.9
200	50.34	10068	167.8

Table 3: Swiping Timings for Typical Event Sizes

Students	Total Processing Time (minutes) Swipe Before Event	Total Processing Time (minutes) Swipe End of Event
50	16.3	8.25
100	33.85	16.5
200	67.7	33

Our solution to this problem was to create and Android mobile application that permits people to "swipe" in using their mobile device. The application contains three functional areas. The first area, shown in Figure 6, instructs a user to enter their first name, last name, id and email address. This is equivalent to the data we gather when we register individuals on the web.

The second area, shown in Figure 7, is where individuals enter and save the names of the events they wish to use the app to swipe in. The user that requests the event tracking provides the event name to an individual. A sample event name is shown in Figure 4, e.g. COMP:1601:A:2014:WINTER.

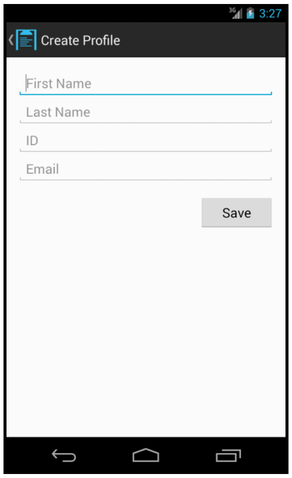
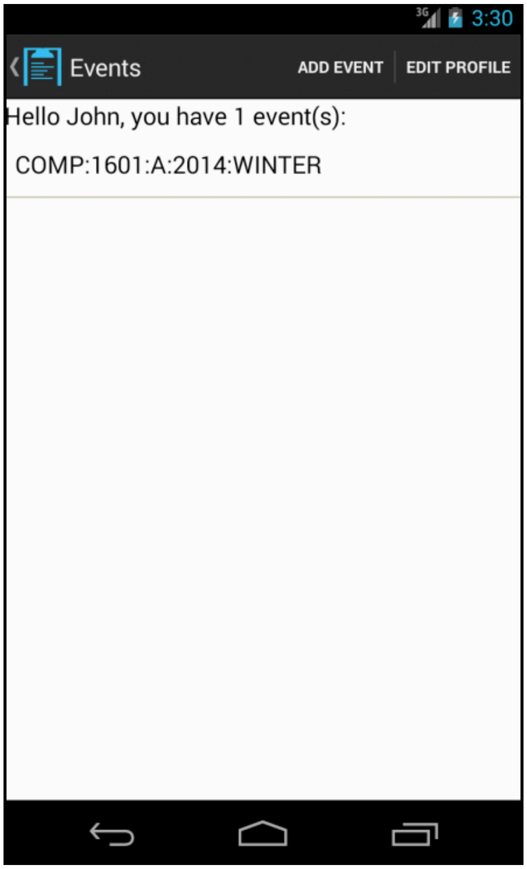


Figure 6: User Data



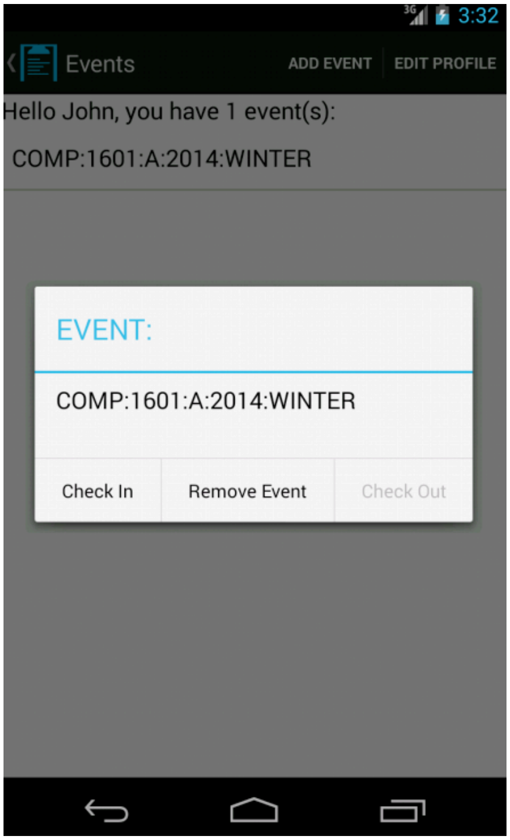
Hello John, you have 1 event(s):

COMP:1601:A:2014:WINTER

Figure 7: Event Identification

The third area, shown in Figure 8, is where an individual signals that they are at the event. This is simply a matter of selecting the event and initiating a check in request to our attendance tracking server. The device's GPS location is sent as part of this message and compared against the GPS location of the actual event to ensure the individual is at the event.

The benefit of this approach is that only the number of simultaneous connections the server can manage limits the number of simultaneous event trackings. Even then the actual processing time is minimal, so classes of 100 and 200 do not create a problem for the system. The disadvantage to this approach is that people require the app on their mobile devices. In the situation where an individual doesn't have a mobile device or doesn't want to install the app, the fall back approach is the manual swiping and registration method.



Hello John, you have 1 event(s):

COMP:1601:A:2014:WINTER

EVENT:

COMP:1601:A:2014:WINTER

Figure 8: Event Swipe

6 References

[1] Deugo D., "Attenance Tracking". The 9th International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 2013), 79-85, 2013.

[2] Durden G. and Ellis L., 1995. The Effects of Attendance on Student Learning in Principles of Economics. The American Economic Review, 85(2), 343-346. Available at <http://www.jstor.org/stable/2117945> [Accessed Jan 15 2014].

[3] http://www.fordham.edu/academics/office_of_research/research_centers__in/center_for_teaching_/teaching_tip_archive/taking_attendance_in_90441.asp, [Accessed Jan 15 2014].

[4] https://support.google.com/drive/answer/87809?hl=en, [Accessed Jan 15 2014].

[5] Heriot Watt University, http://www.hw.ac.uk/registry/resources/studentattendancepolicy.pdf, [Accessed Jan 15 2014].

[6] Minnesota State University, http://www.mnsu.edu/cetl/teachingresources/articles/classattendance.html, [Accessed Jan 15 2014].

[7] Nyamapfene A., Does class attendance still matter?, Engineering Education, Vol 5, 2010.

[8] Reda A., Panjwani S., and Cutrell E., Hyke: a low-cost remote attendance tracking system for developing regions. InNSDR, Washington, D.C., 2011, Jan 15, 2014, http://research.microsoft.com/pubs/156114/nsdr011-reda-Hyke.pdf.

[9] Rodgers J.R., Encouraging tutorial attendance at university did not increase performance, Australian Economic Papers, 41 (3), 255-266, 2002.

[10] Romer D., Do Students Go to Class? Should They?, Journal of Economic Perspectives, Vol 7, Num 3, 167-174, Summer 1993.

[11] http://www.gae.ucm.es/~padilla/extrawork/tracks.html, [Accessed Jan 15 2014].

Computational Quiz Generation

Daniel Hoffman, Felix Giannelia and Ming Lu
Department of Computer Science
University of Victoria
PO Box 3055 STN CSC
Victoria, BC V8W 3P6
Telephone: 250-472-5768
FAX: 250-472-5708
Email: {dhoffman,niteling,luming}@cs.uvic.ca

***Abstract*—Every computer science instructor would like to have quizzes which can be delivered online, with the answers captured and marked automatically. Online quizzes are useful in traditional lecture courses, more useful in online courses and extremely useful in massively online open courses (MOOC). While multiple choice quizzes are already widely used, they are limited. In particular, they are often not effective in evaluating procedural mastery, e.g., tracing a C program. We present *CQG*, a novel online quiz generation framework focusing on evaluation of procedural mastery. The framework includes code and detailed processes for the development of new questions and new question types. The question type development process is based on a reference architecture and a verification procedure aimed at minimizing server crashes and marking errors. *CQG* has been used in first, second and third-year courses. Twenty-three quizzes have been developed, focusing on code tracing in C, C++, Java and Python, on procedures commonly taught in computer networking courses—such as computing a Hamming code—and on the classic encryption and decryption algorithms.**

Keywords-code reading; active learning; web-based tool

I. Introduction

Online quizzes relieve the instructor of the tedium of marking exercises: every instructor's dream. In courses with a traditional lecture format, online quizzes are helpful, especially if the class is large. In courses with online delivery, online quizzes are a perfect fit. We believe that the biggest impediment to the widespread use of massively open online courses (MOOC) is the development of much better online evaluation.

Online quizzes are in use today, primarily in two formats. Multiple choice quizzes are easy to create and deliver and, while limited, are effective for some skill evaluations. At the other extreme are sophisticated online simulations or labs. This approach provides impressive capabilities but is expensive to develop, deliver and maintain.

We present Computational Quiz Generation (*CQG*), a web application framework exploiting the opportunity between multiple choice and online labs/simulations. With this approach, there are four fundamental steps:

1) The instructor selects a computation.
2) The instructor creates a question, in template form, by replacing one or more items with placeholders. We call the placeholders "hotspots" and say that this step "activates the computation."
3) The student fills in the hotspots, producing a completed template.
4) The application automatically checks the student solution by performing the computation on the completed template.

We illustrate these steps with a well-known procedure: adding a column of integers. Figure 1(a) contains a completed addition problem. In Figure 1(b) the two digits in the sum have been replaced by hotspots a and b. The student provides values for a and b; obviously, the solution is correct if and only if $a = 7$ and $b = 0$

This simple example illustrates several important points about the *CQG* procedure:

1) So called "backward questions" are possible, where the hotspots are inputs to the computation. Note that, in a backward question, the correct answer may differ from the original. For example, in Figure 1(c), the solution $c = 6$ and $d = 4$ is also correct.
2) While step 2 guarantees that the question is solvable, many selections of digits and hotspots are not solvable, as shown in Figure 1(d).
3) Step 4 requires answer *checking* not answer *generation*; the latter is often much harder.
4) The complete computation and the templates derived from it form an abstraction hierarchy. Figures 1(b) and (c) are more abstract than Figure 1(a). Figure 1(e) is more abstract than Figures 1(a–d).

CQG is a web application implemented with HTML forms technology. Each question is presented as a template. The student provides answers in HTML input elements, such as text boxes and pull-down lists, and clicks a "check" button. *CQG* checks the completed template and issues a message: "correct" or "incorrect".

Section 2 illustrates *CQG* from the student point of view, with detailed discussion of four questions and three question types. Section 3 explains how to use *CQG* to generate many questions from a single original computation and how to produce reusable quizzes from a large pool of questions. Section 4 presents the main server design goals, and discusses

$$\begin{array}{cc} 2 & 3 \\ 4 & 7 \\ \hline 7 & 0 \end{array}$$

(a) Original

$$\begin{array}{cc} 2 & 3 \\ 4 & 7 \\ \hline a & b \end{array}$$

(b) Traditional

$$\begin{array}{cc} 2 & c \\ 4 & d \\ \hline 7 & 0 \end{array}$$

(c) Backward

$$\begin{array}{cc} 2 & 0 \\ 4 & d \\ \hline 7 & 3 \end{array}$$

(d) Unsolvable

$$\begin{array}{cc} 2 & c \\ 4 & d \\ \hline a & b \end{array}$$

(e) Abstract

Fig. 1. Question generation steps: adding a column of integers

both the server failures encountered to date and the methods developed to eliminate those failures. Section 5 summarizes our experience with *CQG*. Section 6 presents the related work in quiz generation in computer science.

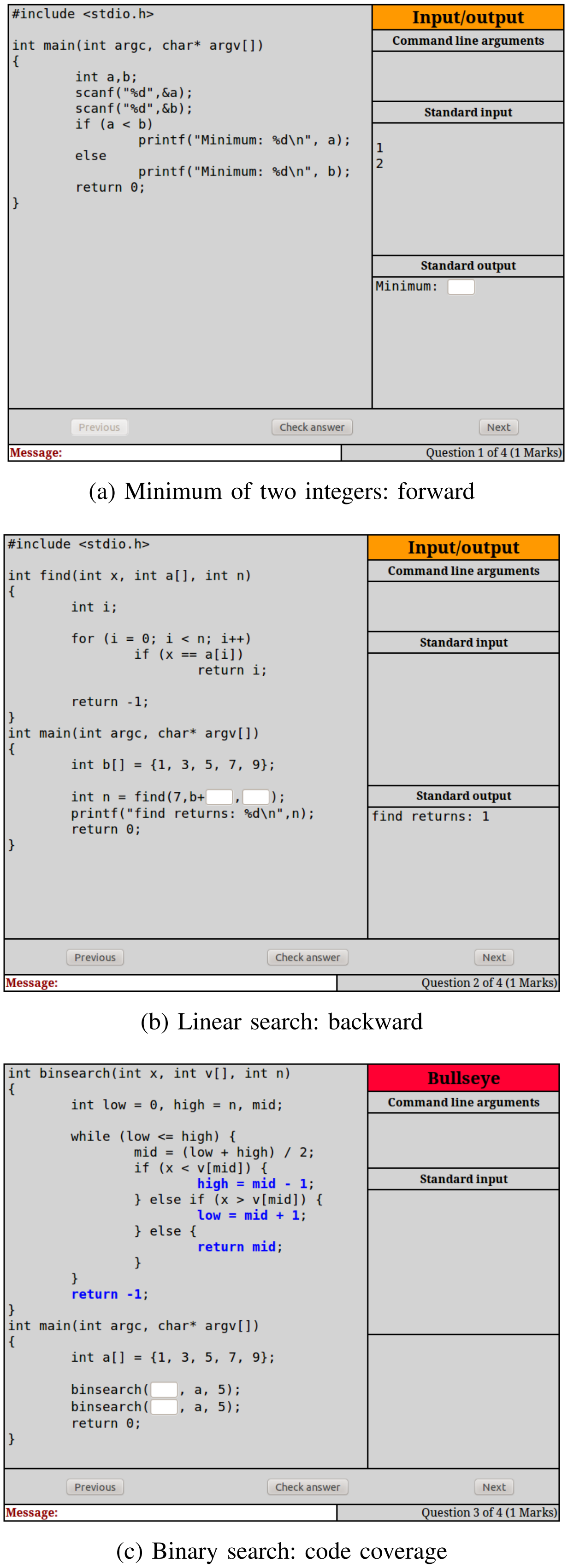
II. Student View

In the *CQG* system, a quiz consists of a list of questions. Each question has a question type; fourteen question types are currently provided.

A. Active code reading

Many instructors believe that students would write better code if they spent more time reading code; many instructors already do ask students to read code. Unfortunately, the reading tends to be unfocused and little is learned. We lack precise goals and practical evaluation mechanisms for code understanding. Many students simply do not know how to learn much about a program by reading it.

Active code reading is a special case of active learning. Figure 2(a) shows a question of type *Input/output*, designed to support active code reading. On the left side is a complete C program; on the right are screen cells for command-line arguments (not used in this question), standard input and standard output. The student must complete the text field in the standard output cell and press Check answer. The answer is correct if the standard output shown on the screen, including the textbox value, is identical to the standard output from an actual execution.



```
#include <stdio.h>

int main(int argc, char* argv[])
{
        int a,b;
        scanf("%d",&a);
        scanf("%d",&b);
        if (a < b)
                printf("Minimum: %d\n", a);
        else
                printf("Minimum: %d\n", b);
        return 0;
}
```

Standard input:

```
1
2
```

Standard output:

```
Minimum: 
```

Message:

Question 1 of 4 (1 Marks)

(a) Minimum of two integers: forward

```
#include <stdio.h>

int find(int x, int a[], int n)
{
        int i;

        for (i = 0; i < n; i++)
                if (x == a[i])
                        return i;

        return -1;
}
int main(int argc, char* argv[])
{
        int b[] = {1, 3, 5, 7, 9};

        int n = find(7,b+   ,   );
        printf("find returns: %d\n",n);
        return 0;
}
```

Standard output:

```
find returns: 1
```

Message:

Question 2 of 4 (1 Marks)

(b) Linear search: backward

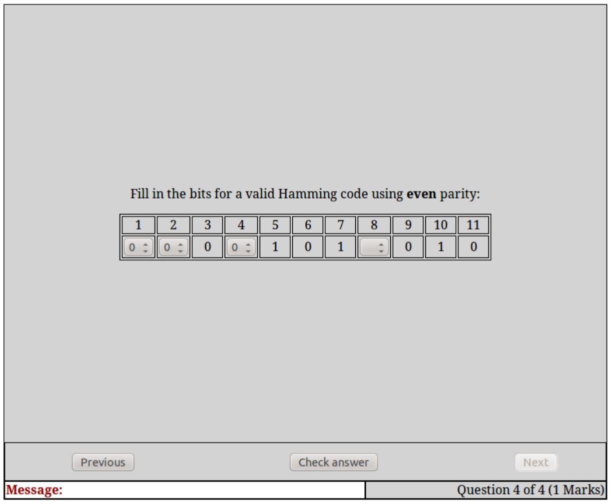
```
int binsearch(int x, int v[], int n)
{
        int low = 0, high = n, mid;

        while (low <= high) {
                mid = (low + high) / 2;
                if (x < v[mid]) {
                        high = mid - 1;
                } else if (x > v[mid]) {
                        low = mid + 1;
                } else {
                        return mid;
                }
        }
        return -1;
}
int main(int argc, char* argv[])
{
        int a[] = {1, 3, 5, 7, 9};

        binsearch(   , a, 5);
        binsearch(   , a, 5);
        return 0;
}
```

Message:

Question 3 of 4 (1 Marks)

(c) Binary search: code coverage

Fig. 2. The *CQG* application: Code Activator question types



Fill in the bits for a valid Hamming code using **even** parity:

1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	1	0	1		0	1	0

Message:

Question 4 of 4 (1 Marks)

Fig. 3. The *CQG* application: Hamming Code question type

Figure 2(b) shows an *Input/output* question focusing on linear search using “slices” of C arrays. In the call to `find`, the student may alter only the displacement d and length n of the slice `b[`$d..d+n-1$`]`. The student is asked to complete an invocation of `find` such that the value 7 is located in position 1 of the slice. In a correct answer $d = 2$ and n is 2 or 3.

Figure 2(c) illustrates the *Bullseye* question type. Again, active code reading is the goal. There are two changes in screen layout from *Input/output*: (1) there is no standard output cell and (2) there are one or more highlighted lines in the code. The student must complete the text fields such that each highlighted line is executed at least once. In Figure 2(c), there are four highlighted lines. The function `main` invokes `binsearch` twice, with a textbox for the target element `x`. The net effect is that the student must choose two `x` values which achieve 100% statement coverage of `binsearch`.

CQG supports a third active code reading question type. With *Find-the-failure*, the code cell contains a specification of a function F, a purposely faulty implementation of F, and an invocation of F. The student must supply failure-revealing values, i.e., values which cause the standard output to be incorrect with respect to the specification.

While C is used in the examples in this section, C++, Java, and Python are also supported.

B. A Hamming code question type

CQG question types need not be based on programming language code, as shown by the *Hamming* question type in Figure 3. In a Hamming codeword, the bits in positions $1, 2, 4, 8, \ldots$ are check bits; the other bits are message bits. The check bits are computed using a generalized parity scheme. In Figure 3, the student has used the pull-down lists to enter the first three of the four check bits. The answer is correct if the completed form is a correct Hamming code.

C. Quiz modes

The tool is designed for two types of quiz: practice and marked.

1) In a practice quiz, the student is informed immediately whether his/her answer is correct. No record is kept of the student’s work.
2) In a marked quiz, the student must log in before beginning the quiz. While no immediate feedback is given regarding the correctness of his/her answers, the answers to each question are logged. Marks are computed automatically from the log.

III. Question author view

Our goal is to develop quizzes which are usable by students side-by-side in a lab and reusable by students term by term. Such quizzes require a large pool of questions. Creating good automated quiz questions is hard, requiring skill and experience, as well as refinement through repeated use and improvement. Many instructors have the ability and desire to develop such questions but do not have enough time unless the questions can be generated.

A. Question generation from templates

Question generation in *CQG* is based on templates. For example, the question shown in Figure 2(a) was generated from the template shown in Figure 4. Generation is implemented in Python; in actual use, the question author assigns the values shown in Figure 4 to Python variables. The C code is separated into global code and main code to make it easier to generate the extra code occasionally needed to handle answer checking. Following the code templates are templates for command-line arguments, standard input and standard output. Each of the five templates can contain a mixture of plain text and markers for hot spots: locations which might contain a text field in a delivered question. In Figure 4, the markers `$x0`, `$x1`, and `$y0` are used; a marker can be any string which does not already occur in the templates.

The list of tuples in Figure 4 specifies how the hot spots are to be filled in: with a string or numeric constant, or with `None`. The latter indicates that the hot spot should appear as a text field. Since there are five entries in the tuple list, this template will generate five questions. The question in Figure 2(a) was generated from the tuple `[1,2,None]`.

The third tuple, `[None,None,1]`, generates a backward question. The standard input contains a textbox for each of `a` and `b` and the standard output (`1`) is provided; the student must supply `a` and `b` values such that the output is `1`. Forward questions test procedural mastery in a straightforward way. With patience and knowledge of C, any forward question can be answered correctly. Backward questions often require more sophisticated thinking. Viewed another way, while *CQG* could generate the answer for any forward *Input/output* question, generating answers for backward questions is an undecidable problem. Because *CQG* checks only completed templates, the undecidability of answer generation is not an issue.

Global code

```
    #include <stdio.h>
```

Main code

```
    int a,b;
    scanf("%d",&a);
    scanf("%d",&b);
    if (a < b)
       printf("Minimum: %d\n", a);
    else
       printf("Minimum: %d\n", a);
```

Command-line arguments

Standard input

```
    $x0
    $x1
```

Standard output

```
    Minimum: $y0
```

Tuple list

```
    [1,2,None]
    [9,3,None]
    [None,None,1]
    [None,None,20]
    [None,None,-2]
```

Fig. 4. Question template for *Input-output*

Figure 4 illustrates one use of the template abstraction hierarchy. Each *CQG* question is a template to be completed by the student. Each question template, such as the one shown in Figure 4, is a more general template, completed partially and repeatedly by the author.

The question in Figure 2(b) was generated from a template with four hotspots: the three parameters to `find` plus the return value in the standard output: `[`x,d,n,y`]`. From that template, the tuple `[7,None,None,1]` generated the question in Figure 2(b). Dozens of four-tuples can be generated with cut-and-paste in a few minutes. For example, in increasing order of difficulty are forward questions (1) with $d = 0$ and $n = 0$, (2) with one of d and n non-zero, and (3) with both d and n non-zero.

The question in Figure 2(c) was generated from a template with six hotspots: one for each of the four target lines and the target elements in each of the two `binsearch` invocations. From that template, the tuple `[True,True,True,True,None,None]` generated the question in Figure 2(c). Easy questions contain only a single targeted line while harder questions include more targeted lines.

The question in Figure 3 was generated from a significantly different kind of template. Here, each template contains one message bitstring, and a list of hotspot indexes for each generated question. The indexes are (a) one-relative and (b) refer to the code word generated from the message bitstring, not the message itself. The forward question in Figure 2(d) was generated from the message `'0101010'` and index list `[1,2,4,8]`. Easy questions are forward and contain very few hotspots. While it is easy to generate a few Hamming Code questions by hand, the template format makes tuple generation attractive. Because the template file is written in Python, arbitrary Python code can be embedded. Consider the Python function `g(`m,n`)` which takes message string m and generates all index lists of length n into the code word. This function is easy to implement in Python, and can be used to generate large numbers of Hamming questions.

B. Quiz specification with random selection

Support is provided for packaging questions in quizzes. The author can simply provide a list of questions, along with a mark for each; each student will see the same quiz. Given a sufficient collection of templates, however, it is easy to create a “cheat-proof” quiz, where copying from a neighbor is no help at all. To accomplish this, the author specifies a list of question groups. Each group consists of a set of pedagogically equivalent questions and a number N, indicating the number of questions to be randomly sampled from the group. Each time the quiz is begun by a student, the sampling is done again.

Consider a quiz constructed from the five questions generated from the template in Figure 4. Suppose that the author chooses two groups: the two “forward” questions and the three “backward” questions. A two-question quiz could be constructed by selecting one question from each of the two groups. Each time a student begins the quiz, he or she will receive one of the six possible quizzes.

CQG automatically collects data on student performance including each submitted answer and the time spent on the question.

IV. Design and verification

A. Design goals

Aside from fast question generation and reasonably fast question type development, there are three interdependent *CQG* design goals:

1) *Minimal by design*. We use no clip art, no JPEG images and no animation. Our intention is to focus the student's attention entirely on the activated computation itself.
2) *Low server load*. A new question retrieved from the server typically contains a few hundred bytes. A client answer submitted to the server is typically under 100 bytes. Answer checking is usually very fast. For example, in a C *Input/output* question, the code is precompiled. The *Hamming* answer checking is very fast for the short bitstrings appropriate for quiz questions.
3) *Secure*. *CQG* has been designed to be secure against attack because it is certain that it will be attacked. HTML forms [1] are used with no client-side embedded

code, e.g., no Javascript. HTML pages are generated on the server side using the web2py [2] framework which provides good support for secure servers.

All answers are type checked to avoid failures during answer checking. For example, in an *Input/output* question, textbox entries are restricted to string and numeric constants, making the answer checking immune to code insertion attacks. *Hamming* questions use pulldown lists offering 0 and 1 only. In both question types, the restrictions are checked server side, to protect against hacking of HTML GET/POST parameters.

B. Question type and question template verification

As in all software projects, *CQG* has seen a number of runtime failures. While the software is not safety-critical, the failures are costly. The 24-by-7 availability of online quizzes is lost; failures during marked quizzes are especially problematic.

We focus on failures due to (1) faults in a question type implementation and (2) faults in a question template. As is common in Python code, question type implementations make extensive use of nested lists and dictionaries. Almost all failures were due to list or dictionary dereferencing errors.

We have attacked question type implementation failures with active code reviews. In each source file, each list or dictionary dereference is considered. The reviewer is asked to prove, informally, that the dereference is guaranteed to be legal. Usually the proof cannot be carried out unless certain preconditions are present. In practice, the proofs are usually short and simple. Most of the effort is consumed by writing and maintaining the preconditions.

Most template failures appeared as questions which were either unsafe for one or more answers, or unsolvable. For example, the question shown in Figure 2(b) is unsafe. Because array `b` has 5 elements, if a student enters, e.g., `100` in the first textbox, the program will attempt to reference addresses far outside the memory allocated for `b`. As a result, the behaviour of the code is unpredictable. On the other hand, if the author changes the standard output value to, e.g., `2`, then the question is unsolvable. In simple templates with just a few tuples, the generated questions are usually safe and solvable. In complex questions with a lot of tuples, we often see a few unsafe or unsolvable questions.

To ensure that questions are solvable, we usually use the approach illustrated in Figure 1: each question is derived from a solved computation by adding hotspots. While it is relatively easy to avoid unsolvable questions, unsafe questions are problematic. In particular, for each question type, a process for demonstrating question safety must be developed. In the case of C *Input/output* questions, the template author must prove, typically manually, that any generated question will be safe. Recall that the question in Figure 2(b) is unsafe because entries in the first (slice displacement) or second (slice length) textbox may cause dereferencing errors. The template author can avoid these problems by (1) activating only one of the displacement and length fields in any given question and (2) using a list box rather than a text box to restrict the displacement and length to safe values. Of course, with C++, Java, or Python questions, exception handlers can be used to achieve safety more elegantly.

With the *Hamming* question type, safety is guaranteed. The generator checks that the message is a non-empty bitstring and that the indexes are in range. Answers are provided using listboxes offering only bit values, making every generated question safe.

V. Classroom experience

CQG has been used in four different courses at the University of Victoria:

1) *CSc 111—Fundamentals of Programming wih Engineering Examples.*
2) *SEng 265—Software Development Methods.*
3) *SEng 360—Security Engineering.*
4) *CSc 361—Computer Communications and Networks.*

For *CSc 111* and *SEng 265*, there are eight quizzes based on the active code reading question types using C,covering the following topics:

1) *Loops.* The first quiz covers `for`, `while`, `break`, and `continue`.
2) *Parameter passing.* Many of the students have a Java background, where the language forces call-by-reference for objects and call-by-value for primitives. Consequently, they struggle with the use of `*` and `&`.
3) *Pointers and arrays.* Even for experienced programmers, there are misunderstandings on these topics.
4) *String libraries.* This quiz covers the use and sample implementations of the most popular functions.
5) *Linked lists.* The standard insertion, deletion, and search algorithms are covered.

In *CSc 111*, the quizzes are distributed across the term; In *SEng 265*, they are completed in the first four weeks.

For *SEng 265*, there are also four quizzes based on the active code reading question types using Python:

1) *Lists.* List creation, including list comprehension and list slices, are covered.
2) *Control structures.* The standard constructs are covered, focusing on looping over objects supporting iteration, such as lists.
3) *Functional features.* The `filter`, `map`, and `reduce` functions are covered, as well as lambda expressions.
4) *Regular expressions.* There are two quizzes on regular expressions, covering (1) simple matching and (2) string parsing.

In *SEng 265*, for example, the same approach scheme was used for each quiz. In each quiz, there were 20 questions; each question was selected randomly from a group with at least 25 questions. The quiz was made available for practice on Monday. Students took the quiz as many times as they liked. In the lab on Thursday the quiz was taken for a mark, with some additional questions and a 30-minute time limit. In the most recent offering of *SEng 265*, students were required to develop their own *CQG* questions, focusing on n^2 sorting algorithms in C.

In *SEng 360*, there are five quizzes covering the Caesar, Substitution, Columnar Transposition, Vernam, Book, and RSA (with very small keys) encryption and decryption algorithms. Cryptanalysis using letter, digram and trigram frequencies is also covered.

In *CSc 361*, in addition to the *Hamming* question type shown in Figure 2(d), five question types were developed for the Fall 2012 offering:

1) *CRC*. The cyclic redundancy check procedure used for error detection in Ethernet packets is covered. This procedure uses a specialized binary long division.
2) *FDB*. The algorithm used to implement packet forwarding in Ethernet switches is covered.
3) *Dijkstra*. Knowledge of Dijkstra's shortest path algorithm is evaluated. No code is shown on the display. Instead, the student must provide the intermediate shortest path results for each iteration of the algorithm.
4) *IP address*. A pair of IP addresses and a subnet mask are provided; the student must determine whether the addresses are on the same subnet.
5) *NAT*. The algorithm used to implement the network address translation used in home routers is covered.

The *CSc 361* question types show the variety of procedural skills which can be evaluated using *CQG*. In five previous offerings, these procedures were presented in lecture and then evaluated on exams. In the Fall 2012 offering, after *CQG* practice, the students performed substantially better on the exams.

VI. Related work

Code Activator, a *CQG* predecessor, offered just the three active code reading applications: *Input/output*, *Bullseye*, and *Find-the-failure* [3], [4]. *CQG* improves on *Code Activator* in three important ways:

1) After using the *Code Activator* application, we realized the potential for a generalization: from individual product to product line. We significantly redesigned the system to make question type a plugin and implemented the six computer network question types.
2) We developed the question and the question type verification processes.
3) We used *CQG* in three more offerings of *SENG 265* and in *CSC 361*, developing many new questions.

The psychology literature contains reports that active retrieval is a more effective learning strategy than simple repetition [5]. Industrial software reviews have been shown to benefit from a question-based, active-learning approach [6]. Active learning has long been advocated in the Computer Science education community [7], [8]. A variety of approaches have been used to engage students. Liu has shown how to improve learning outcomes by replacing traditional lectures with a 50/50 split between lectures and question-based work sheets [9]. Wu's castle [10] is a graphical role-playing game for teaching students about the execution behaviour of loops. Students can set various loop parameters, e.g., index start and end values. To retain student interest, execution is carried out by animated characters. With PeerWise [11], a web-based tool for active learning, students collaboratively create multiple choice questions. Use of the tool correlates positively with later exam performance. Peer instruction [12], long used in physics instruction, was adapted to programming courses in Java. Clickers are used in lecture to answer questions focusing on previously assigned readings and exercises.

Some approaches to active learning incorporate automated checking of student work. GraphPad [13] is a web-based tool aimed at graph data structures. Students draw graphs which are compared to the instructor's solution automatically using a graph isomorphism algorithm. The tool also captures student interactions at the pen-stroke level for later study by the instructor. Kumar [14] presents a web-based tutor aimed at teaching C++ pointers. The tutor presents C++ code with a pointer error; the student must identify the variable and line number of the error. Both practice and marked mode are supported. The tutor is part of the problets framework [15].

With ProgTest [16], A student implements a program according to a specification. The program's correctness is automatically evaluated by running it against unit tests provided by the instructor.

VII. Conclusions

In lecture-based, online and MOOC courses, it is very useful to have online quizzes. *CQG* is a novel framework for developing online quizzes using template-based questions to evaluate procedural mastery. The *CGQ* framework includes detailed development processes for questions and for question types. Fourteen question types have been developed to date. *CQG* quizzes have been effective in evaluating procedural mastery in active code reading, encryption/decryption algorithms, and a variety of computer networking tasks.

Acknowledgment

The authors would like to thank H. Wang and H. Lien for help with authoring *C-doku* questions and the students of SEng 265 for their patience with bugs in the *C-doku* server.

References

[1] P. Carey, *New Perspectives on HTML, XHTML, and Dynamic HTML*, 4th ed. Cengage Learning, 2010.

[2] "The official web2py book," 2010, http://www.web2py.com/book.

[3] D. Hoffman and M. Lu, "A web tool for active code reading," in *Proc. FECS 2010*, 2010.

[4] D. Hoffman, M. Lu, and T. Pelton, "A web-based generation and delivery system for active code reading," in *Proc. SIGCSE*, 2011.

[5] E. Jaffe, "Will that be on the test?" *Assoc. Psych. Science*, vol. 21, no. 10, pp. 18–21, 2008.

[6] D. Parnas and D. Weiss, "Active design reviews," in *Proc. of the Intl. Conf. on Software Engineering*, 1985.

[7] J. McConnell, "Active learning and its use in Computer Science," in *Proc. SIGCSE/SIGCUE Conf. on Integrating Technology into Computer Science Education*, 1996.

[8] J. Whittington, "Increasing student retention and satisfaction in IT introductory programming courses using active learning," in *Proc. 2006 Informing Science and IT Education Joint Conf.*, 2006.

[9] Z. Liu, "Is paper-based work sheet out of date for CSII?" in *Proc. FECS 2009*, 2009.

[10] M. Eagle and T. Barnes, "Experimental evaluation of an educational game for improved learning in introductory computing," *SIGCSE Bull.*, vol. 41, no. 1, pp. 321–325, 2009.

[11] P. Denny, B. Hanks, and B. Simon, "Peerwise: replication study of a student-collaborative self-testing web service in a U.S. setting," in *SIGCSE '10: Proc. of the 41st ACM technical symposium on Computer Science education*, 2010, pp. 421–425.

[12] B. Simon, M. Kohanfars, L. Michael, J. Lee, K. Tamayo, and Q. Cutts, "Experience report: peer instruction in introductory computing," in *SIGCSE '10: Proc. of the 41st ACM technical symposium on Computer science education*. ACM, 2010, pp. 341–345.

[13] R. P. Pargas and S. Bryfczynski, "Using ink to expose students' thought processes in CS2/CS7," *SIGCSE Bull.*, vol. 41, no. 1, pp. 168–172, 2009.

[14] A. N. Kumar, "Data space animation for learning the semantics of C++ pointers," *SIGCSE Bull.*, vol. 41, no. 1, pp. 499–503, 2009.

[15] "Problets: the home page," 2010, http://www.problets.org.

[16] D. de Souza, J. Maldonado, and E. Barbosa, "Progtest: An environment for the submission and evaluation of programming assignments based on testing activities," in *Proc. CSEE&T*, 2011.

Development of a Computer-Based Instructional Design tool - Analysis and Design Phases

Bénédicte Talon[12], **Christophe Kolski**[13], **and Mouldi Sagar**[14]
[1] Université de Lille Nord de France
[2]LISIC, Université du Littoral Côte d'Opale, Calais, France
[3]LAMIH, UVHC / CNRS, UMR 8201,Valenciennes, France
[4]DeVisu, UVHC, Valenciennes, France

Abstract - *This paper discusses the computerization of an existent tool called GRASP. It helps teachers to design an instructional system. Currently, GRASP has the form of a grid. Teachers complete it to first clarify their instructional problem and then to design an instructional system. This tool covers the analysis and design phases of the ADDIE (Analyze, Design, Develop, Implement and Evaluate) model. GRASP has been validated in a two-phase evaluation. The paper discusses the pertinence and the adequate way to realize the computerization of GRASP; the proposed system is called CB-GRASP. For this, the paper begins with the context of the work. Then a review literature informs about instructional design. GRASP and its use are then described. The analysis and design phases of CB-GRASP are presented. The perspectives of this work conclude this paper.*

Keywords: Instructional Design; Instructional Design method, Instructional Design tool

1 Introduction

This paper is in the field of Instructional Design (ID). A non-computerized tool named GRASP (GRid bASed Pedagogical design) that assists teachers during analysis and design phases of an Instructional System has been created; this tool is detailed in [1]. Teachers answer to successive questions in order to clarify their instructional problem and design a pedagogical environment.

To provide an example, this tool has enabled the design of an instructional system for teaching databases, intended for undergraduate learners in a short cycle class [1]. With the help of GRASP, the teacher has refined the instructional problem (e.g., develop skills in designing 3NF databases while working on transversal competences such as planning, organization and resource management). This teacher has produced a collaborative instructional system to develop the skills needed on the job and the transversal competences needed to remedy the deficiencies in project management.

The tool (in its textual form) has been evaluated on a public composed of 15 teachers [1]. The results show its ability to help teachers to develop pedagogical devices tailored to their professionalization goals and demonstrate implementation of more active learning activities. However, during the evaluation phase, several teachers underlined the importance and the interest for this tool to be computerized. The paper describes the work done in this direction.

This paper reviews the terminology related to ID. It presents briefly GRASP and its evaluation. The issues of this tool in a computerized form (called CB-GRASP) are then discussed. The conclusion discusses the opened perspectives.

2 The Instructional Design Domain: Literature review

2.1 Instructional Design and Instructional System

Instructional Design (ID) is a method or process used to produce plans and models describing the organization of learning and teaching activities, resources and actors involvement that compose an Instructional System or a Learning Environment [2]. Many authors summarize ID as a *systematic planning of instruction* [3].

Instruction is the intentional facilitation of learning towards identified learning goals [4]. The Instructional System is a complex system, either computerized or not, designed to manage an instructional problem. It combines organizational conditions, arrangement of resources and procedures allowing the instruction to take place. The instructional problem is a set of variables concerning the learner, the teacher and the environment (such as the learner's profile, the instruction domain, the teacher's expertise, his/her preferences, the number of learners, the size of the room, the pedagogical material, and so on) that will affect the learning process. Figure 1 gives a pictorial vision of these concepts.

2.2 Theories, Models and Methods in ID

It is not easy to structure the ID world. In fact, literature of ID is full of terms sometimes used as synonyms due to misuse of language.

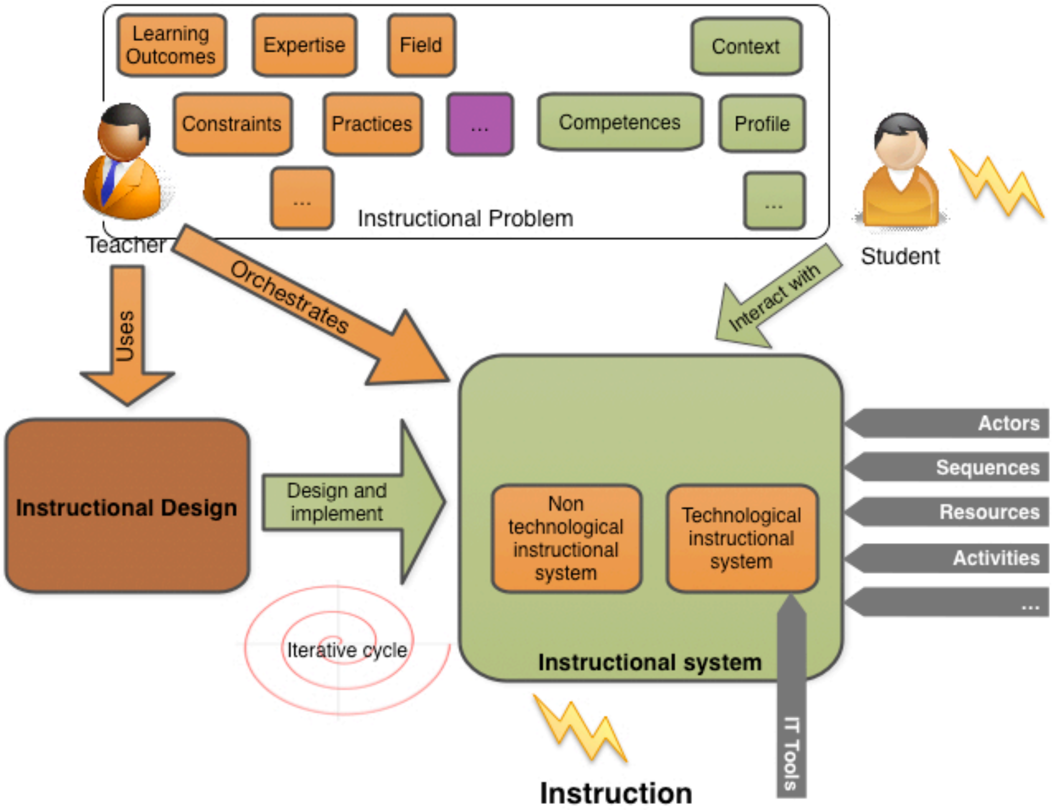


Fig. 1. Situating Instructional design

A review of literature has allowed synthesizing this complex domain. The concepts are described in the following paragraphs and pictured this into figure 2.

A designer uses an **Instructional Design Method** to organize the design of an instructional system in order to solve an ID problem. The ID Method defines a process, models and/or tools used during this process.

The ID method applies an **Instructional Design Model**, which proposes a structure and a meaning to a specific ID problem. According to [5] an ID model provides a procedural framework for the systematic production of instruction and incorporates elements of the ID process and may be used in different contexts. It prescribes how combination of instructional strategy components should be integrated to produce a course of instruction. According to [6] "*the role of models in Instructional Development is to: provide us with conceptual and communication tools that we can use to visualize, direct and manage processes for generating episodes of guided learning; allow us to view both the linear and concurrent aspects of instructional development; and to allow us to select or develop appropriate operational tools*".

An ID model refers to an **ID theory**. The theory explains phenomena, focuses on the ways to attain learning goals and "*offers explicit guidance on how to better help people learn and develop*" [7].

Instructional methods are principles and methods used for instruction. They rest on an **instructional model**. The choice of an instructional method depends on the ID theory.

Learning theories inform on the process of how human learn. Learning theories consider the variables that influence the learning process and provide explanations on how this influence occurs. Learning theory serves as the bridge between research and education [8]. It helps to define educational frameworks, which define the principles through which the theory can be applied to learning and teaching practices.

2.3 Instructional Designers' Practices and Tools

Several authors [9,10,11,12,5] have examined how ID designers work. Observations show that many designers do not work in the linear process proposed by common ID models.

Approaches marrying prototyping, zigzagging, top-down and bottom-up design are commonly practiced and instructional designers exploit problem solving and decision-making skills [13].

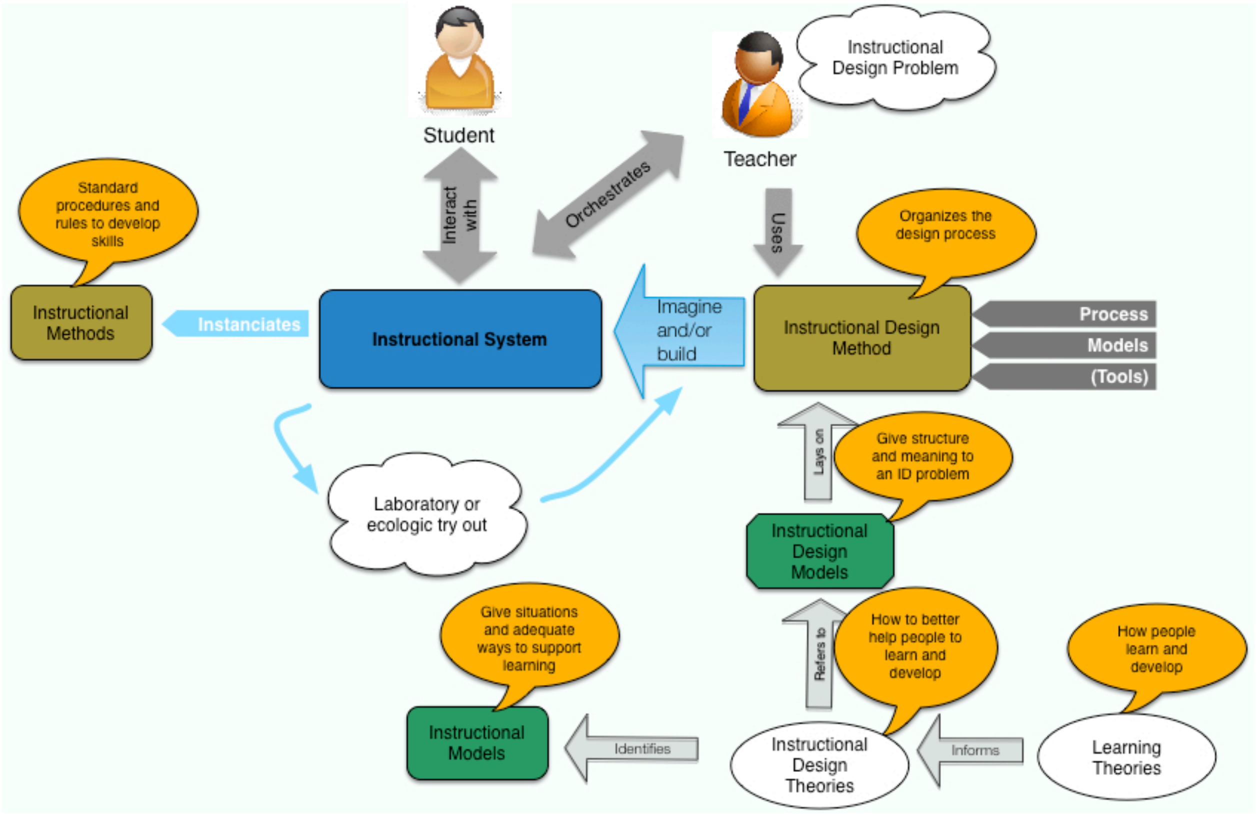

Student
Teacher
Instructional Design Problem
Standard procedures and rules to develop skills
Instructional Methods
Instanciates
Instructional System
Interact with
Orchestrates
Uses
Organizes the design process
Imagine and/or build
Instructional Design Method
Process
Models
(Tools)
Lays on
Give structure and meaning to an ID problem
Laboratory or ecologic try out
Instructional Design Models
Give situations and adequate ways to support learning
Refers to
How to better help people to learn and develop
How people learn and develop
Instructional Models
Identifies
Instructional Design Theories
Informs
Learning Theories

Fig. 2. Concepts of Instructional Design

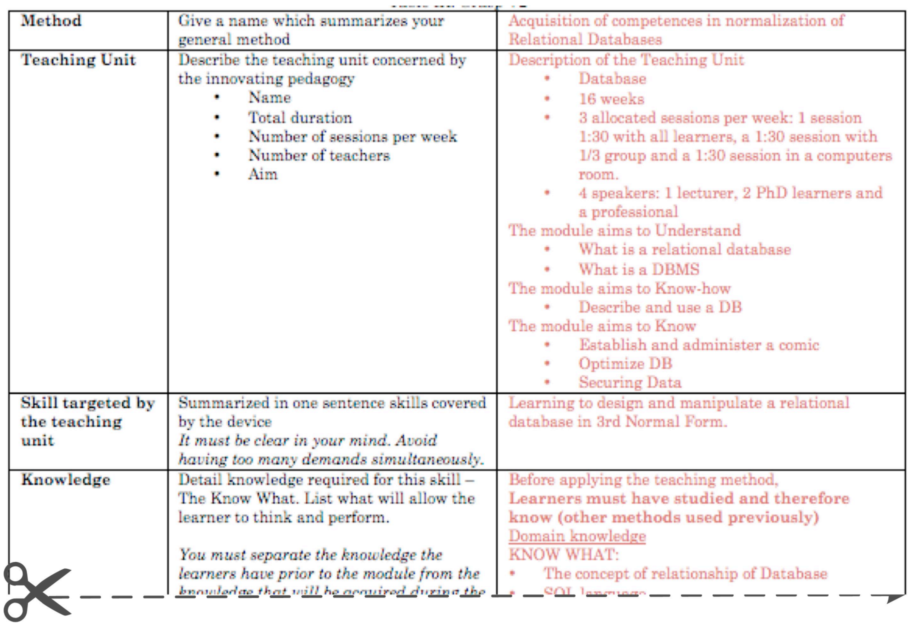
Rowland [11] observes that while designers refer to ID model by applying some suggested activities they often omit one or more activities for reasons such as lack of time and the fact that they are considered as unnecessary. According to [14] designers use three different approaches to design and implement instructional systems: (1) designs based on instructional theory, (2) designs based on best practices and (3) designs based on patterns in best practices. Concerning the use of tools to support the design process, they are used at different moments and under different forms. Van Merriënboer and Martens [12] describe a range of ID tools: (1) pre-authoring systems that support needs assessment or the whole analysis and design process before a final medium selection is made, (2) systems for the selection of methods and media and (3) systems for supporting the implementation and evaluation of instructional systems after the instruction has been developed.

2.4 Instructional Design Models

An **Instructional Design Model** provides a procedural framework for the systematic production of instruction. It incorporates elements of the ID process and may be used in different contexts [5]. The model prescribes how combinations of instructional strategy components should be integrated to produce instruction.

A model of Instructional Design focuses on different components and strategies [3]. Needs analysis and strategic planning are still one of the most important areas of ID and Instructional Systems development today. In 1980, Andrews and Goodson [15] already announced more than 40 ID models. The most often cited ID model is ADDIE (Analyze, Design, Develop, Implement and Evaluate) even if according to Molenda [16], the ADDIE model is rather to be seen as a family of procedural and systematic ID models.

A constructive approach of ID views it as the preparation of resources and learning processes in order to facilitate students learning in a constructivist approach. Rather than emphasizing predetermined design steps, the constructive approach focuses on the iterative development and implementation of learning environments offering opportunities for inquiry and discovery learning [3]. The family of constructivist instructional models offers an approach closely related to complex problem solving.



Method	Give a name which summarizes your general method	Acquisition of competences in normalization of Relational Databases
Teaching Unit	Describe the teaching unit concerned by the innovating pedagogy • Name • Total duration • Number of sessions per week • Number of teachers • Aim	Description of the Teaching Unit • Database • 16 weeks • 3 allocated sessions per week: 1 session 1:30 with all learners, a 1:30 session with 1/3 group and a 1:30 session in a computers room. • 4 speakers: 1 lecturer, 2 PhD learners and a professional The module aims to Understand • What is a relational database • What is a DBMS The module aims to Know-how • Describe and use a DB The module aims to Know • Establish and administer a comic • Optimize DB • Securing Data
Skill targeted by the teaching unit	Summarized in one sentence skills covered by the device *It must be clear in your mind. Avoid having too many demands simultaneously.*	Learning to design and manipulate a relational database in 3rd Normal Form.
Knowledge	Detail knowledge required for this skill – The Know What. List what will allow the learner to think and perform. *You must separate the knowledge the learners have prior to the module from the knowledge that will be acquired during the*	Before applying the teaching method, **Learners must have studied and therefore know (other methods used previously)** Domain knowledge KNOW WHAT: • The concept of relationship of Database • SQL language

Fig. 3. Extract from GRASP 2.0

It shares the approach with the areas of architecture, design engineering, graphic design, and other fields of design with a larger place to creativity [17,18]. This family favors rapid prototyping [19] after a succinct needs analysis.

3 The GRASP tool (textual version)

The GRASP tool offers a support to designers in the instructional design process for the development of innovative pedagogical environments. The work was based on the observation that the teachers are ready to change their systems but need help and do not want to perform a too long process. The design of the tool has followed a prototypal approach in 4 steps (1) stating the problem defining the context and aim, (2) inspecting learning theories and the ID domain, (3) developing a tool (in a textual version) and (4) evaluating it. An iterative application of these steps results in the production of the GRASP V1.0 and GRASP 2.0 grids and their evaluation. GRASP 1.0 was evaluated by three teachers. The aim was to test the feasibility of using this tool in an academic environment. Another evaluation tested the validity and usability of GRASP V2.0 with 15 teachers from various fields. An evaluator was at the side of each teacher to answer possible questions and to make choices explicit according to the explicitation interview [20]. The detail of these evaluations modalities is available in [1].

3.1 GRASP 2.0 (second version)

GRASP is a grid divided in three parts. Teachers must fill successive cells to answer questions. The instructional system is designed through three main steps (the three parts of the grid). The first part focuses on target skills (the Intended Learning Outcomes). In the second part teachers define their educational context and constraints. The third part is dedicated to the design choices: role of the teacher, resources for the learners, type of process, etc. Teachers build their teaching scenario. Here follows an extract from GRASP 2.0 (figure 3 - in red an example concerning an instructional system dedicated to database management learning that can help the teacher)

3.2 Evaluation Results and opened perspectives

The evaluations [1] showed that GRASP interests teachers and that they don't notice difficulties in using this tool. Teachers have designed instructional systems implementing more active forms of pedagogies and have become aware of the need to assess differently.

An interesting point of the evaluation was the discovery that the interaction between the teacher and the evaluator has promoted confrontation of ideas. This confrontation was favorable to questioning the old methods and moving towards a more innovative education, which better fits the needs of students and teachers.

Several teachers asked when they should use a computerized version of the tool. But this computerization may take various forms: from a simple grid to an intelligent assistance tool. The question is what form of assistance would be the best. This paper tries to give elements to answer this question. To design and implement the tool we are currently applying a prototypal design process. It first implies an analysis and design of the computer-based tool, called CB-GRASP (Computer-Based GRid bASed Pedagogical design).

4 Toward a computerized version of GRASP

It is usual to establish a definition of requirements during the specification phase of a project. The content of this document structure the activities by defining clear orientations for the work to be done. In order to be clear with the finalities of the project, this section presents the specific context of the work obtained from different evaluations and literature.

4.1 Requirements Analysis

The aim of the project is to produce a computer-based GRASP tool to help teachers/designers during the analysis and design of instructional systems.

4.1.1 Scope of the tool

During evaluations of GRASP, teachers explained that they essentially need assistance during the analysis and design phase. The analysis is the process of gathering data to identify specific needs — the *Who, What, Where, When, and Why* of the design process. The design Phase takes the information compiled from the analysis phase and builds an instructional system definition. The tool concentrates on this problematic and consequently, the development, implementation and evaluation phases are not the scope of our current work.

As Van Merrinboër et al. [12] noted, "*The field of ID tools has been dominated, and still is dominated, by authoring tools for the development or production of computer-based instruction… But until now, there has been less interest in computer-based ID tools that support the analysis and design activities that are undertaken before a final medium selection is made and the instruction, which may be not computer-based, is actually produced*".

The tool concentrates on the analysis and design phases of the instructional design lifecycle.

4.1.2 Instructional design process

The analysis and design process must reflect natural practices of designers. These practices depend from individuals and experience. Studies show that, often, the designers do not act according to a linear process. Many designers explore the solution while specifying the problem and expert designers realize cycles of try-out and improvement [10]. For example, constraints are rarely identified during the first analysis step but are rather introduced at every step of the process [21].

We want to favor cyclical approaches allowing designers to navigate in the tool according to their internal process of creativity. However, the tool should offer best practices to guide them.

4.1.3 Personalization

Gustafson [21] notes that it is important that the tool could be configured to be adapted to user needs. The acceptance of a tool is correlated to its usability, which depends from the fact that we are taking the user's profile and practices into consideration.

The tool will offer the ability to adapt to different design approaches.

4.1.4 Assistance modalities

The evaluation showed that designers appreciate to interact with the evaluator to solve their instructional problem. This interaction has fostered the understanding of their problem and has favored their creativity. The aim is not to develop an authoring tool or an expert system but to produce good advices thanks to relevant questioning**.**

We want to develop a tool that offers mechanisms to foster the creativity of designers. The tool must ask relevant questions and offer advices in accordance with their problem.

4.1.5 Instructional problem

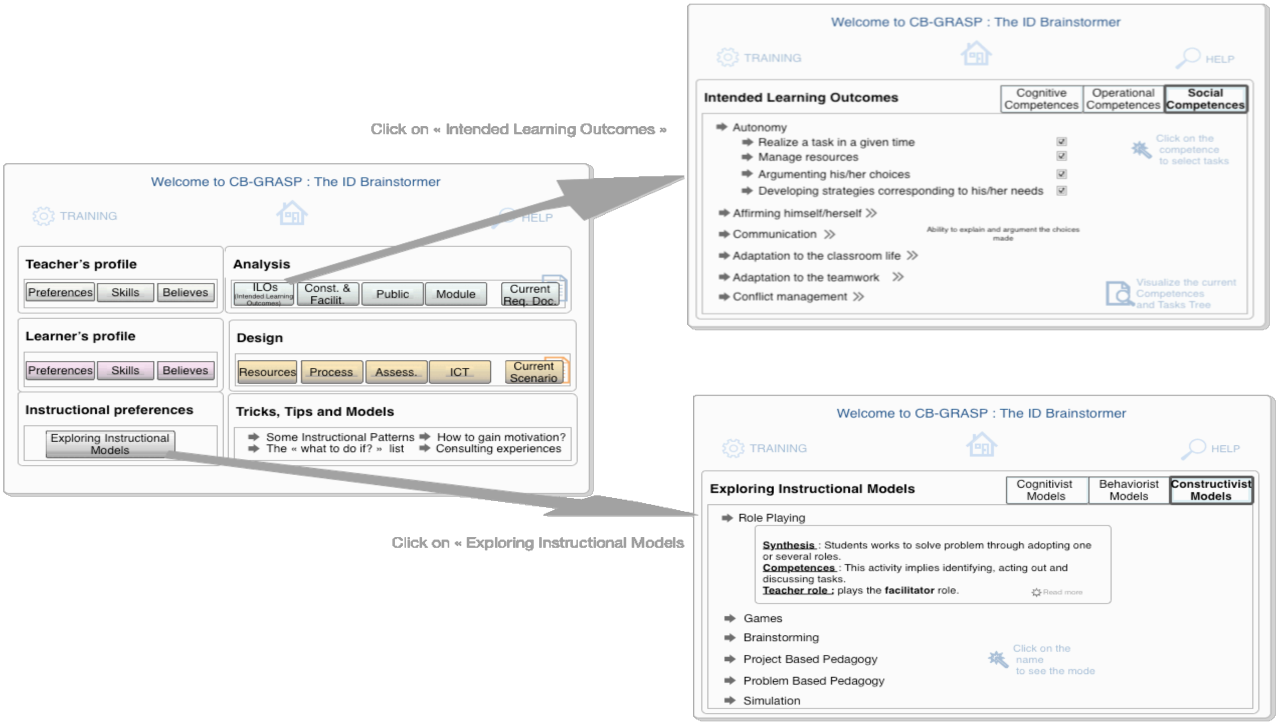
GRASP has been built according to the constructive alignment theory [22]. This theory favors in-depth approaches in which pedagogy should focus on critical thought, dialogue and discussion. Thus, the training situation should suggest questions to students, who, by finding the responses, acquire additional experience and new skills. Teachers must create situations that force them to carry out a task, produce something or solve a problem. Some instructional methods can be found in [1,23].*We will favor social constructivist theories but other theories are acceptable.*

4.2 Design phase

4.2.1 Mock-ups realization

The design phase started with the realization of mock-ups respecting the requirement identified in the analysis phase. They are designed in a real will to respect each teacher's design process.

Some mock-ups (figure 4) illustrate the philosophy, notably the free navigation in the tool. The homepage has been designed in order to allow a free navigation. Designers can navigate between the different activities according to their preferences and even refine their choices.



Click on « Intended Learning Outcomes »

Welcome to CB-GRASP : The ID Brainstormer

TRAINING HELP

Teacher's profile

Preferences | Skills | Believes

Analysis

ILOs (Intended Learning Outcomes) | Const. & Facilit. | Public | Module | Current Req. Doc.

Learner's profile

Preferences | Skills | Believes

Design

Resources | Process | Assess. | ICT | Current Scenario

Instructional preferences

Exploring Instructional Models

Tricks, Tips and Models

- Some Instructional Patterns
- The « what to do if? » list
- How to gain motivation?
- Consulting experiences

Welcome to CB-GRASP : The ID Brainstormer

TRAINING HELP

Intended Learning Outcomes

Cognitive Competences | Operational Competences | **Social Competences**

- Autonomy
  - Realize a task in a given time ☑
  - Manage resources ☑
  - Argumenting his/her choices ☑
  - Developing strategies corresponding to his/her needs ☑
- Affirming himself/herself »
- Communication »
- Adaptation to the classroom life »
- Adaptation to the teamwork »
- Conflict management »

Ability to explain and argument the choices made

Click on the competence to select tasks

Visualize the current Competences and Tasks Tree

Welcome to CB-GRASP : The ID Brainstormer

TRAINING HELP

Exploring Instructional Models

Cognitivist Models | Behaviorist Models | **Constructivist Models**

- Role Playing

  Synthesis : Students works to solve problem through adopting one or several roles.
  Competences : This activity implies identifying, acting out and discussing tasks.
  Teacher role : plays the **facilitator** role.
  Read more

- Games
- Brainstorming
- Project Based Pedagogy
- Problem Based Pedagogy
- Simulation

Click on the name to see the mode

Click on « Exploring Instructional Models

Fig. 4. Some CB-GRASP Mock-ups

On the main page, the "Help" button offers information on the tool functionalities. The "Training" button gives access to a page that offers explanations about the design approach and the tool. It describes each button and the navigation modalities in each window. The "ILOs" button leads to a page that offers three tabs: each tab corresponds to a type of know-how (operational, cognitive and social know-how). In each tab, teachers can navigate in a competences model. They can select specific know-hows they want to develop. Each competence is linked to a set of tasks that are likely to develop this competence. Teachers can input know-hows that are not provided in the tool. This may be the case of operational know-hows that are specific to the instructional field.

All selections and data entering are stored. The teachers can click on the current requirement document or the current scenario to download a document that synthetizes all elements that have been already informed.

The teacher's profile and learner's profile definitions help to structure the learning context. Each button of these profile definitions leads to a page where designers navigate in trees and select elements.

The "Tricks, tips and models" space allows the elaboration of solutions after an exploration of these elements. The "Exploring instructional models" page helps teachers to imagine and build the instructional system. During the exploration, designers can select items and visualize their specificities.

4.2.2 Mock-ups evaluation

The current phase of work concerns the evaluation of the mocks-ups in order to improve the effectiveness of their design. There is a strong desire that the evaluation begins in the early stages of the tool production. An iterative design model is preferred through a practice of a user-centered design. The approach emphasizes analysis of teachers' tasks and activities in order to reflect how they actually work. So, all actions and comments made by the teachers about mock-ups will be observed and analyzed.

In order to validate acceptation by teachers, a focus will be made on the concept of usability. The ability of the tool to really assist the teacher during the implementation of an instructional environment will be evaluated. The main question is 'is it relevant for teachers to easily achieve their objectives?'. The evaluation must so insist on the quality of interface (ergonomics): structuring, information presentation, etc. The questions are: 'Is the design environment easy to use?', 'Is it easy to learn?', 'Is it comfortable?', 'Is it flexible?', 'Is it error-free?'.

In order not to miss any functionalities of the system, a focus will be made on the concept of utility. The assessment should ensure that the tool really checks the specifications described in section 4.1.

In order to improve the ergonomics of the tool, we will implement an already tried and tested method. It was practiced by one of the authors who is specialized in software ergonomics for the modernization of a control room and has involved field operators. The aim of this method is to exploit users' needs to correct and build the final mock-ups. This method will concern teachers (the same teachers as those who were implied in the evaluation of GRASP 2.0). These 15 teachers will define, according to a specific process, the layout of the tool that would help them to develop an IT-supported learning environment. The activity that will be asked to them will be: "*imagine and draw the tool that should facilitate your design work*". Designers will have access to the GRASP grid version. They will also have access to models of mock-ups (of unrelated fields). Production time will be limited to one hour. There will be two groups of teachers. A group will work after a pre-visualization of the designed mock-ups (figure 4). Another group will work without this support. We will compare the results and will cross them with questionnaires filled in by the designers before and after the work on mock-ups.

The results of this work will be merged in order to produce the second version of mock-ups. Then, a second human-centered evaluation will be done before to undertake the production of the operational tool. During the implementation of the tool, traditional methods of quality control will be implemented and then, before the distribution of CB-GRASP, user satisfaction studies and critical incident detection will be performed.

5 Conclusion and Perspectives

In this paper, we have presented the instructional design domain and the GRASP tool. We propose to produce a computer-supported tool to better help teachers to design pedagogical environments adapted to their context. For this purpose, we have defined the requirements and elaborated mock-ups to highlight the principles of CB-GRASP, a computerized version of GRASP. The current phase is the mock-ups evaluation phase. 15 teachers are implicated in the evaluation-design cycle of suited mock-ups. The next phases of our work will be the architecture definition and the iterative development of the computerized tool; they will be described in other papers.

6 References

[1] Talon B., Sagar M., & Kolski C. (2012). Developing Competence in Interactive Systems: The GRASP tool for the design or redesign of pedagogical ICT devices. *ACM Transactions on Computing Education*, 12, 1-43.

[2] Tracey, M. W., Richey, R. C. (2007). ID model construction and validation: a multiple intelligences case Education. *Tech Research Dev*, 55:369–390. DOI 10.1007/s11423-006-9015-4.

[3] Zierer, K., Seel, N.M. (2012). General Didactics and Instructional Design: eyes like twins. *A transatlantic dialogue about similarities and differences, about the past and the future of two sciences of learning and teaching*. 1:15. SpringerPlus

[4] Smith, P. L., & Ragan, T. J. (1999). *Instructional design* (p. 3). New York, NY: Wiley.

[5] Braxton, S., Bronico, K., & Looms, T. (1995). *Instructional design methodologies and techniques*. The George Washington University in Washington.

[6] Gustafson, K., Branch R.M. (1997). Revisioning models of instructional development. Educational Technology Research and Development, 45, 3.

[7] Reigeluth, C. M. (1999). Instructional-design theories and models: A new paradigm of instructional theory. Volume II, Mahwah, NJ: Lawrence Erlbaum Associates (Ed.).

[8] Schunk, D. H. (2012). *Learning theories – An educational perspective*. Pearson. 6th Edition. ISBN-13: 978-0-13-707195-1.

[9] Merrill, M. D., Wilson, B. (2007). *The Future of Instructional Design. Trends and Issues in Instructional Technology (2nd ed)*. R. Reiser and J. V. Dempsey (Eds.), Upper Saddle River NJ: Pearson Prentice Hall, 335-351.

[10] Visscher-Voerman, J. (1999). *Design Approaches in Training and Education: A Reconstructive Study*. Doctoral Dissertation, University of Twente, Enschede, The Netherlands.

[11] Rowland, G. (1992). What do designers actually do? An initial investigation of expert practice. Performance Improvement Quarterly, *5*(2), 65–86.

[12] Van Merriënboer, J.J.G., Martens, R. (2002). Computer-Based Tools for Instructional Design: An Introduction to the Special Issue. *Educational Technology Research and Development*, *50*(4), 5–9.

[13] Visscher-Voerman, J. (1999). *Design Approaches in Training and Education: A Reconstructive Study*. Doctoral Dissertation, University of Twente, Enschede, The Netherlands.

[14] Brouns, F., Koper, R., Manderveld, J., Van Bruggen, J., Sloep, P.,Van Rosmalen, P., Tattersall, C., & Vogten, H. (2005). A first exploration of an inductive analysis approach for detecting learning design patterns. *Journal of Interactive Media in Education*, Vol. 3, 1-10.

[15] Andrews, D.H., Goodson, L.A. (1980). A Comparative Analysis of Models of Instructional Design. *The Journal of Instructional Development, 3*(4), 2-16.

[16] Molenda, M. (2003). *The ADDIE Model. in Educational Technology: An Encyclopedia*. A. Kovalchick and K. Dawson (Eds), ABC-Clio, Santa Barbara, CA.

[17] Goel, V. (1995). *Sketches of thought*. MIT Press, Cambridge (MA).

[18] Hanke, U., Ifenthaler, D., & Seel N.M. (2011). Effects of creative dispositions on the design of lessons. *The Open Educational Journal,* 4 (suppl. 1:M10),113–119.

[19] Tripp, S., & Bichelmeyer,B (1990). Rapid prototyping: An alternative instructional design strategy, *Educational Technology Research and Development*, 38 (1), 31-44.

[20] Maurel, M. (2009). The explicitation interview: examples and applications, in ten years of viewing from within. *Journal of Consciousness Studies*, 16 (10-12), 58-59

[21] Gustafson, K. (2002). Instructional Design Tools: A Critique and Projections for the Future. *Educational Technology Research and Development, 50*(4), 59–66.

[22] Biggs, J. B. (1999). What the learner does: teaching for enhanced learning. *Higher Education Research and Development, 18*(1), 55-75.

[23] Kolski, C., Loslever, P., & Sagar, M. (2011). The performance of future designers on the specification of supervisory HCI: case study of a simulated work situation. *Cognition Technology & Work*, *14*(2), pp. 107-128.

Prototype Development of Integrated Class Assistance Application Using Smart Phone

Kazuya Murata, Takayuki Fujimoto
Graduate School of Engineering, Toyo University
Kujirai 2100, Kawagoe-City, Saitama Japan
s46d01300037@toyo.jp, me@fujimotokyo.com

Abstract - *Recently, Smart phone spread mainly on 20s. The current smart phone has very high functionality at the same level as a mobile computer. We think that it has high demand to use the high-performance smart phone for learning assistance and class assistance. In addition, we think that the development of the education assistance system has very high demand. However, the smart phone use by the class causes "Inappropriate use" of the student. For this reason, the full-scale research and development do not yet exist. In this research pushes forward development of "Integrative Education Environmental System" to use a smart phone by a class while limiting "Inappropriate use" of the smart phone. In this paper, we design the contents of application for classes and we development prototype system in "Integrative Education Environmental System".*

Keywords: Smart phone; Class assistance; Integrative Education Environmental System; application

1 Introduction

In late years, a cellular phone is the most familiar electronic device. The diffusion rate continues increasing now. In this background, there is the development of cellular phone called a "Smart phone".

The smart phone is development on the basis of the function of a computer. The smart phone has a personal digital assistant function as a big characteristic and enables operation by using touch panel. In addition, the recent smart phone has very high functionality at the same level as a mobile computer, and convenience is very high. For these reasons, the smart phone spreads mainly on businessman and university student. We thought that there was very high demand and functionality for the development of education assistance systems such as universities using smart phone.

However, there is a very big problem with education assistance using smart phone. It is a problem of "Inappropriate use". "Inappropriate use" is network game or "Twitter" that are unrelated to a class. In other words the class use of the smart phone causes "Inappropriate use" using the function of the smart phone. In addition, when we assume a class using a smart phone, all students use a smart phone. Out of these, it is very difficult that a teacher finds "Inappropriate use" and warns. From these reasons, it is very difficult for the class use of the smart phone to realize it. Furthermore, the education assistance system limiting "Inappropriate use" does not yet exist.

2 Purpose

Recently, the realization of the education assistance using smart phone is very difficult. As a cause, there is a problem of "Inappropriate use". Therefore, in this research, we aim at the development of "Integrative Education Environmental System". "Integrative Education Environmental System" is an environmental system enabling the class use of the smart phone while limiting "Inappropriate use".

In this research, we build "Integrative Education Environmental System" by three following structure.

(1) Limit of the cellular radio wave of the smart phone

(2) The wireless LAN that set connection restrictions of the class-limited

(3) Development of the class contents of application

In this structure (1), we limit the cellular radio wave of the smart phone in the classroom. In this structure, we assume that we use a device limiting a cellular radio wave. By this device, we limit the cellular radio wave state of the smart phone. In other words, we limits "communication using a cellular radio wave".

In this structure (2), we use the wireless LAN that set connection restrictions of the class-limited. In this structure, we place wireless LAN only for the class use in the classroom. The smart phone usually has two communication methods, "Communication method using a cellular radio wave" and "Communication method using WiFi function". In the situation that "Communication method using a cellular radio wave" is confined to by structure of (1), "Communication method using WiFi function" is possible. In this research, we use this structure. Furthermore, the wireless LAN connection of the class-limited uses a proxy server. By proxy server, we make only an address and domain to use by a class accessible. With this structure, we build environment "The communication using cellular radio wave is impossible. The student can be connected to the wireless LAN place in the classroom by WiFi function. However, what can be connected only as for the address and domain of the class use".

In structure (3), we develop the contents of application for classes. In this research, we build environment "The communication using cellular radio wave is impossible. The student can be connected to the wireless LAN installed in the classroom by WiFi function. However, what can be connected only as for the address or domain of the class use" by structure (1) and structure (2). Next, we implement contents of application for classes in "Address for exclusive use of the class use". In addition, we develop the contents of application for classes of this research as web application. By this method, we solve a problem to occur from the difference in Operating System of the smart phone.

In these structures, we build "Integrative Education Environmental System" enabling education assistance using a smart phone while limiting "Inappropriate use".

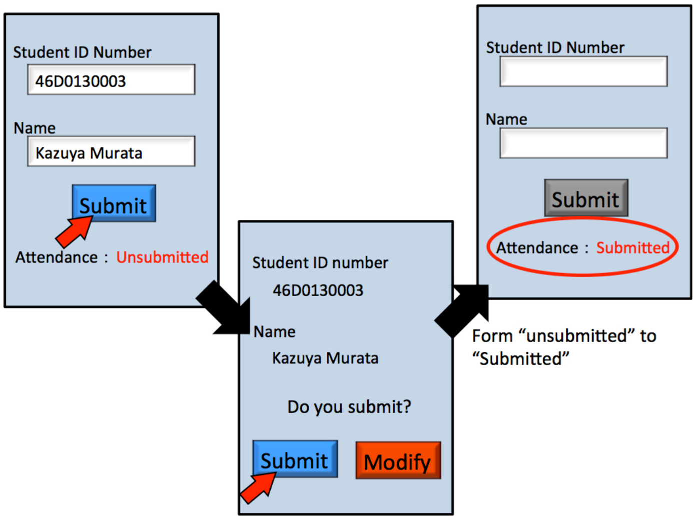
Under the present conditions, we can build the environment to limit "Inappropriate use". However, the contents of application for classes have only a minimum system. Therefore, in this paper, we design the interface of the contents of application for classes to use it by "Integrative Education Environmental System". And we develop a prototype system of the contents of application for classes.

3 Design of the Contents Application for Classes

"Integrative Education Environmental System" of this research, we aim for use enabling a class only by a smart phone and system in the future. Therefore we think that various functions are necessary to manage a class. So we design the interface of the contents of application for classes. In addition, we develop a prototype system of the contents of application for classes. Furthermore, in this research, we are referring to various researches, in order to develop an application system required for class. [1][2][3][4][5][6]

3.1 Transmission of the Attendance Information

By the general class, the confirmation of the student attendance is necessary. For example, in general class, there are usually methods to distribute exclusive paper to for confirmation of the student attendance. In this research, we think about a method to attract the states of the student attendance from application with a smart phone. In addition, the confirmation of the student attendance has a problem. It is a problem of the "substitute attendance". As a method to prevent substitute attendance, this application assumes the transmission of the attendance only once. Therefore, this application displays a page to confirm information when a student transmits attendance information. By this method, we prevent false transmission of the attendance information and substitute attendance. We show a summary of this interface that we designed in figure 1.



Student ID Number
46D0130003
Name
Kazuya Murata
Submit
Attendance：Unsubmitted

Student ID number
46D0130003
Name
Kazuya Murata
Do you submit?
Submit Modify

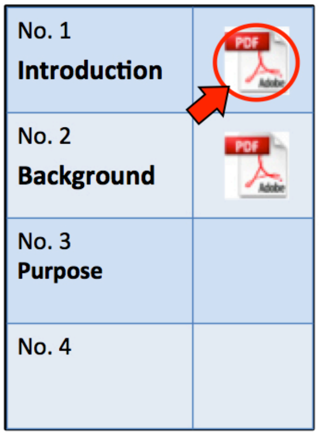
Student ID Number
Name
Submit
Attendance：Submitted

Form "unsubmitted" to "Submitted"

Figure 1. summary of "Transmission of the attendance information"

3.2 Acquisition of the Teaching Materials

In a class, one of important factor is teaching materials. In the case of general class, there are textbook and print document that a teacher made teaching materials. In late years, the PowerPoint document that a teacher made is mainstream. This application system makes it possible to push forward a class smoothly by acquiring the teaching materials from a smart phone. In other words, student can use the smart phone as a textbook. This interface sets up the electronic teaching materials of the Portable Document Format (PDF) file. We think about the interface that student tap PDF icon can download the teaching materials. We show a summary of this interface in figure 2.



No. 1
Introduction

No. 2
Background

No. 3
Purpose

No. 4

Tap of PDF icon and downloaded

Figure 2. Summary of "Acquisition of the teaching materials"

3.3 Contribution and Share of the Question

In a recent class of Japan, there is very few "Remark of the student" that student asks a teacher a question during class. However, it is not a meaning that there is not a question that there are few remarks of the student. Rather we think that there are many students "I want to remark, but cannot do it". Therefore we thought about the application system that a teacher could easily the question of the class. In addition, anyone can look the past questions and answers. By this application system, we enable very smooth class use. We show a summary of "Contribution of the question" in figure 3, and we show a summary of "Share of the question and answer" in figure 4.

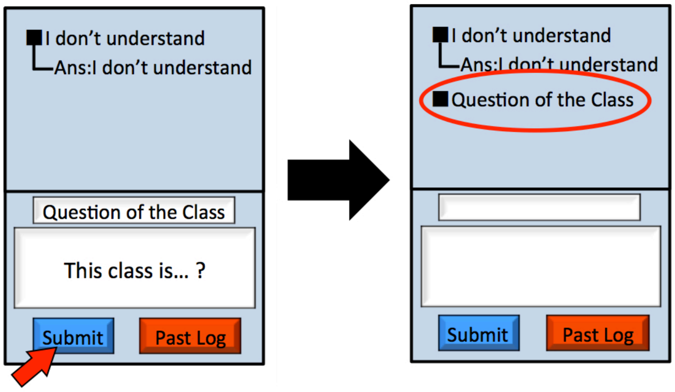


Figure 3. Summary of "Contribution of the question"

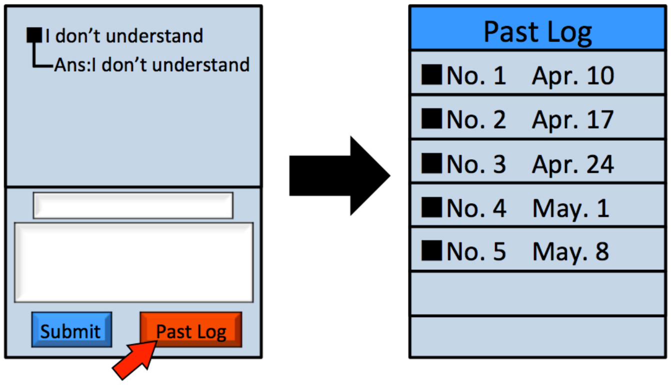


Figure 4. Summary of "Share of the question"

3.4 Question and Questionnaire form a Teacher

When we assume a general class, there is a scene "Question from a student to a teacher" and "Question from a teacher to a student". In the scene of "Question form a teacher to a student", there are very few students answering the question of the teacher. We thought about the application system that student could easily answer by such a question and questionnaire. By this application system, we enable smoother class use. In this interface, we assume it the specifications that the student taps "Yes" and "No" button and can easily answer. We show a summary of this application system in figure 5.

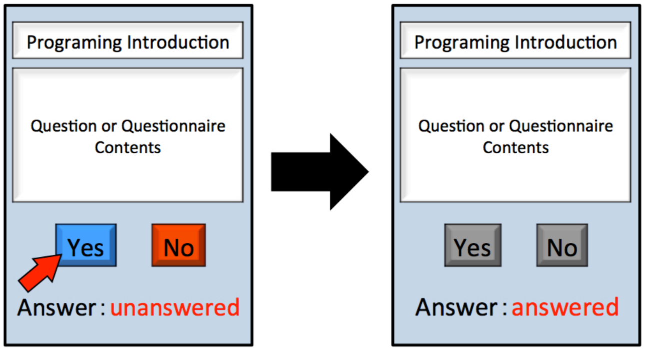


Figure 5. Summary of "Question and questionnaire from a teacher"

3.5 Submission of the Subject

In the class, there is situation to submit a simple subject to at the time of the class end. We thought about the application system that can submit such a simple subject to a student using smart phone. By this application system, we can simplify the collection of the answer to subject. In addition, we can store the answer to subject as electronic data. We show a summary of this application system in figure 6.

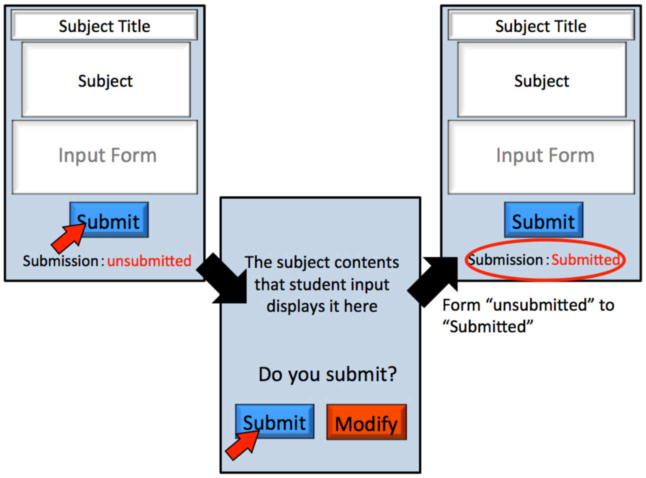


Figure 6. Summary of "Submission of the subject"

3.6 Discussion Space

At the current Japanese university, a student often surpasses 100 students for one class. Therefore it is very difficult to carry out "active learning" in the class. At the Japanese university, it is difficult to carry out a discussion and debate by a large-scale class. For this reason, the class form that a teacher holds is common now. By this application system, with "Integrative Education Environmental System", even a large-scale class enables large-scale active learning using smart phone. In this application system, we think about the function such as the chat to enable the discussion between students. We show a summary of this interface in figure 7.

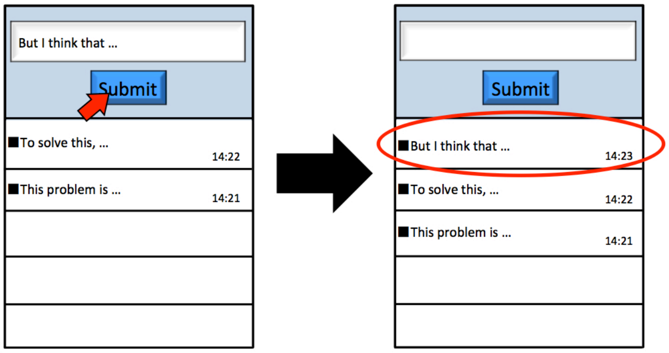
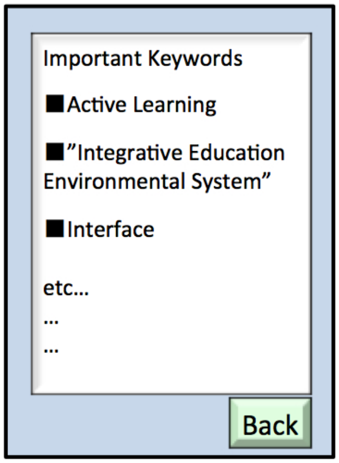


Figure 7. Summary of “Discussion space”

3.7 Explanation of the Teacher

We performed a very simple subject of an experiment in “Integrative Education Environmental System”. In the case of this subject of an experiment, there was an opinion from a student “I concentrated on the operation of the smart phone and have missed the explanation of the teacher”. In reference to such an opinion, we thought that a function to list the explanation of the class was necessary. We show a summary of this interface in figure 8.



Important Keywords

■Active Learning

■”Integrative Education Environmental System”

■Interface

etc...
...
...

Back

Figure 8. Summary of “Explanation of the teacher”

3.8 Glossary

In “Integrative Education Environmental System” to develop in this research, we use a method to limit the communication environment of the smart phone, and to limit “Inappropriate use”. Therefore, student cannot be ”I check that I do not understand it on the Internet”. For example, there is the risk such as “I don’t understand the meaning of a word and term. But I’m not checked”. Therefore we thought that glossary about the class was necessary. We show the interface of this function in figure 9.

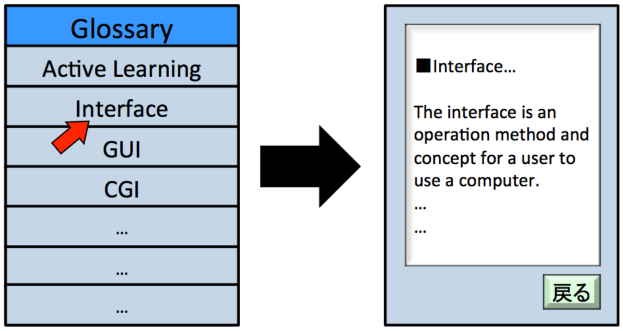

Glossary
Active Learning
Interface
GUI
CGI
...
...
...
■Interface...
The interface is an operation method and concept for a user to use a computer.
...
...
戻る

Figure 9. Summary of “Glossary”

4 Development Prototype of the Contents of Application for Classes

In this paper, we develop a prototype application of the contents of application for classes based on the interface that we designed. In this chapter, we describe an operation example to use prototype application.

4.1 Transmission of the Attendance Information

The transmission of the student attendance needs the following operation.

1. Input a student id number and a full name into an input form
2. Tap a transmission button and move to the confirmation page
3. Tap a transmission button once again

By this operation, the student can transmit the student attendance. We show this application in figure 10.

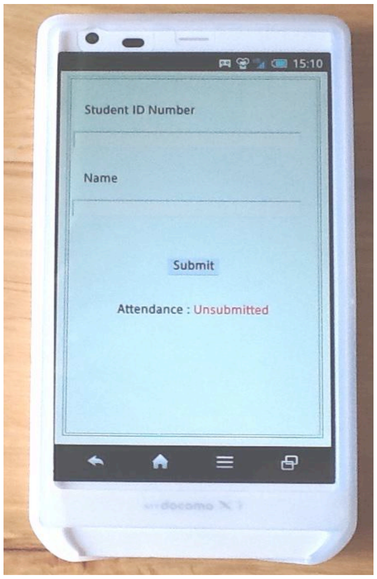


Figure 10. “Transmission of the attendance information”

4.2 Acquisition of the Teaching Materials

The acquisition of the teaching materials needs the following operation.

1. Confirm that subtitle of the class and the PDF icon are installed
2. The teaching materials tap the PDF icon of the class
3. The downloading of the teaching materials is started

By this operation, the student can acquire the teaching materials of the class using smart phone. We show this application in figure 11.

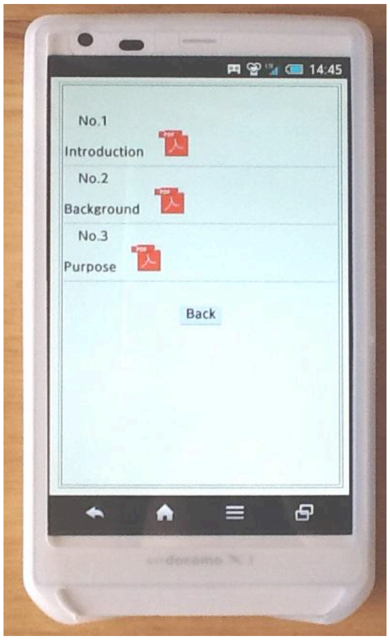


Figure 11. “Acquisition of the teaching materials”

4.3 Contribution and Share of the Question

At first, the contribution of the question needs the following operation.

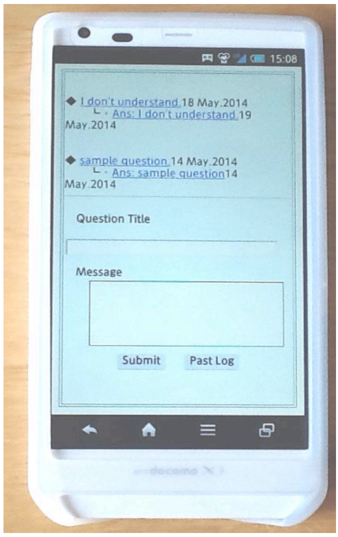
1. Input a question
2. The teaching materials tap the PDF icon of the class
3. The downloading of the teaching materials is started

By this operation, student can contribute the question for the class. We show this application in figure 12.

Next, the share of the question needs the following operation.

1. Tap the “Past Log” button of the lower part
2. Because the date of the class is displayed, student tap the date when student want to read it
3. Because a past question and answer are displayed, student tap it and read it

By this operation, anyone can read a past questions and answers. We show this application in figure 13.


15:08
◆ I don't understand 18 May.2014
└・Ans: I don't understand 19 May.2014
◆ sample question 14 May.2014
└・Ans: sample question14 May.2014
Question Title
Message
Submit
Past Log

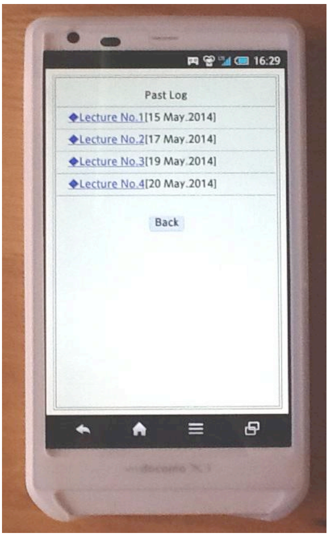

16:29
Past Log
◆Lecture No.1[15 May.2014]
◆Lecture No.2[17 May.2014]
◆Lecture No.3[19 May.2014]
◆Lecture No.4[20 May.2014]
Back

Figure 12. “Contribution of the question”(Left)

Figure 13. “Share of the question”(Right)

4.4 Question and Questionnaire From a Teacher

Question and Questionnaire from a teacher need the following operation

1. Because a question and questionnaire from a teacher are displayed by the center of page, student confirm contents
2. Because a “Yes” and “No” button are installed in the lower part, student tap a button of thinking that it is a correct answer

By this operation, the student can easily answer it by a question and questionnaire from a teacher. We show this application in figure 14.

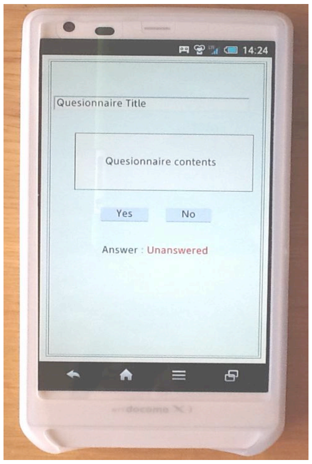


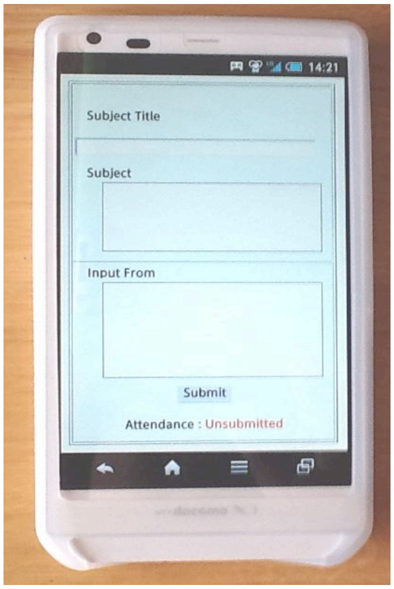
Figure 14. “Question and questionnaire form a teacher”

4.5 Submission of the Subject

The submission of the subject needs the following operation.

1. Because the contents of the subject are displayed by the upper part, student confirm it
2. Student input the answer to subject into an input form installed in the center
3. Student tap “Submit” button of the lower part
4. Because a confirmation screen is displayed, student tap “Submit” button once again

By this operation, the student can submit a subject using smart phone. We show this application in figure 15.



Subject Title

Subject

Input From

Submit

Attendance : Unsubmitted

Figure 15. “Submission of the subject”

4.6 Discussion Space

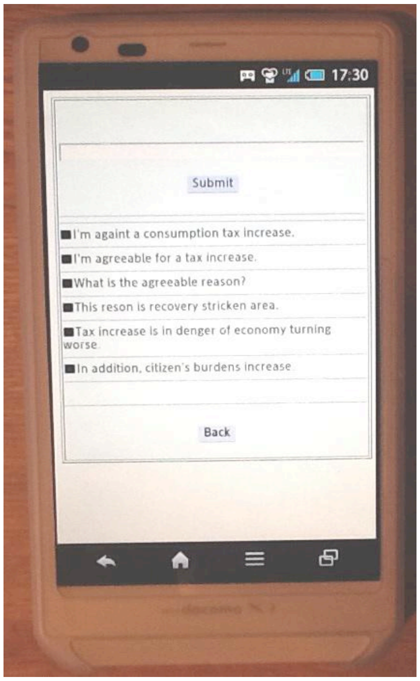
The discussion space needs the following operation.

1. Student input the remark that student want to write in at an input form of the upper part
2. A remark in added to the discussion space of the lower part when student tap a “Submit” button

By this operation, a large-scale discussion using a smart phone is possible. We show this application in figure 16.

4.7 Explanation of the Teacher

By the explanation of the teacher, the explanation of the teacher is listed in the whole page. The student can evade the problem that misses the explanation of the teacher by watching this explanation. We show this application in figure 17.



Submit

■I'm againt a consumption tax increase.

■I'm agreeable for a tax increase.

■What is the agreeable reason?

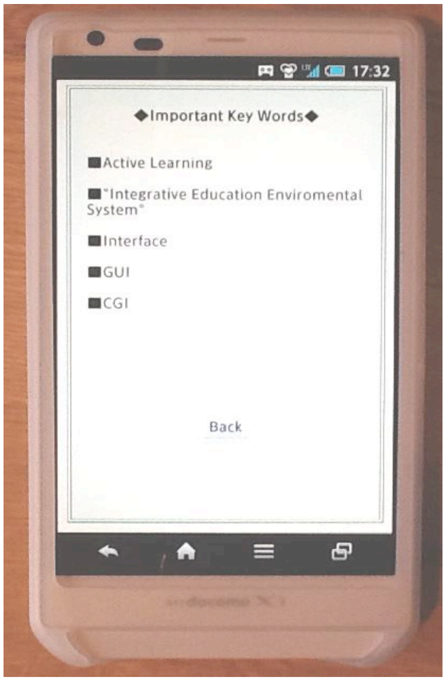
■This reson is recovery stricken area.

■Tax increase is in denger of economy turning worse.

■In addition, citizen's burdens increase

Back

Figure 16. “Discussion space”



◆Important Key Words◆

■Active Learning

■"Integrative Education Enviromental System"

■Interface

■GUI

■CGI

Back

Figure 17. “Explanation of the teacher”

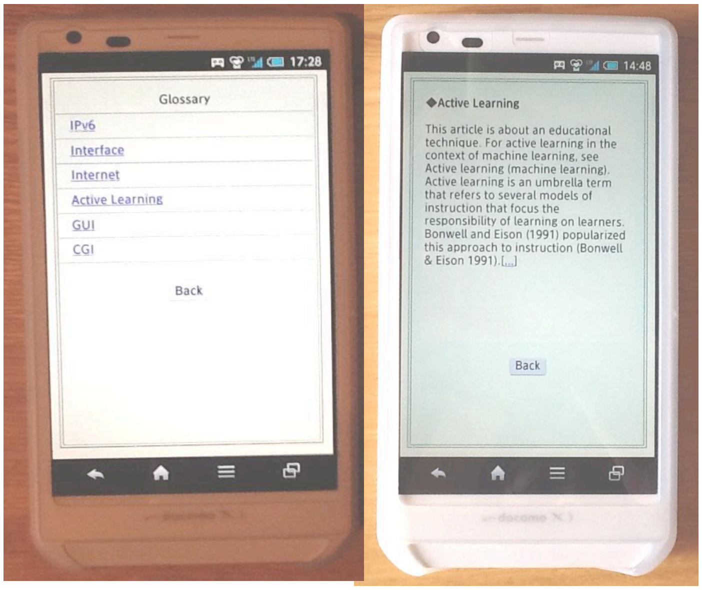
4.8 Glossary

The glossary needs the following operation.

1. On the page, a term connected with a class is display with a list
2. When student tap a term, application move to the page where the meaning of the term is listed in

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

By this operation, student can confirm the meaning of the term of the class without using the Internet. We show this application in figure 18.



◆Active Learning

This article is about an educational technique. For active learning in the context of machine learning, see Active learning (machine learning). Active learning is an umbrella term that refers to several models of instruction that focus the responsibility of learning on learners. Bonwell and Eison (1991) popularized this approach to instruction (Bonwell & Eison 1991).[...]

Figure 18. “Glossary”

5 Conclusion

In this research, it is construction of “Integrative Education Environmental System” that enables class use on the smart phone while limiting “Inappropriate use” as the final aim. In this paper, as one of the important structure of “Integrative Education Environmental System”, we developed a design of the contents of application for classes and prototype application.

However, we only developed prototype application now. So, in the future of this research, we want to push forward improvement based on this prototype application.

As the future of this research, as first we think about the enforcement of the subject of an experiment. As an experiment method, we really have a student use the prototype application that we developed in this paper by a class. Thereafter, we collect opinions from a student and perform addition and deletion of the function of the application. We repeat such a subject of an experiment, we want to complete the application that it is easy to use for student. In addition, with application, we want to aim at the completion of “Integrative Education Environmental System”.

6 References

[1] T. TACHINO, S. KATO, Y.KATO, “University Student’s Use of and Attitude towards Mobile Phones as Part of their Class Activities”, Research report of JET Conferences, 13(1), 247-250, 2013-03-02

[2] M. EBIHARA, “Trial use of cellular phone for a lecture”, Annual Report of Ishikawa Prefectural University: Production, Environment, Food, Based on Biotechnology 22, 33-38, 2011-03-31

[3] N. HONDA, “Development and Trial of an Attendance Management System Using Cell Phones with QR Code”, Otemae Journal 12, 253-262, 2011

[4] T. YAMAMOTO, “Class development using cell phone and web”, Collected treatises, Hamamatsu Gakuin University Junior College, (7), 1-6, 2011-03

[5] M. SHINOHARA, H. SHIMENO, S. IORIO, H. TANAKA, “ET2009-119 Development of Real Time Class Support System by Cell Phone”, Technical Report of IECIE. ET, Educational Technology 109(453), 89-94, 2010-02-26

[6] M. NAKAYAMA, Y. MORIMOTO, K. AKAHORI, Y. SHIMIZU, “Development of the Lecture Support System by using a Mobile Phone”, Technical Report of IECIE. ET, Educational Technology 101(706), 81-86, 2002-03-01

SESSION

ASSESSMENT STRATEGIES + EVALUATION, ANALYSIS, VERIFICATION METHODS + SYSTEMS

Chair(s)

TBA

Assessing Teacher's Teaching Quality in Higher Education: a Preliminary Study

C.Y. Chen[1], **P.C. Chen**[2]
[1]Dept. of IM, National Central University, Jhongli City, Tao-Yuan, Taiwan
cychen@mgt.ncu.edu.tw
[2]Dept. of IM, National Taiwan University, Taipei City, Taiwan

Abstract - *This study presents a preliminary framework that is designed to address teachers' teaching quality (TQ) in terms of teaching capability and maturity. The framework is developed for the assessment to be based on T-CMM. The process design of the framework applies the concepts of SCAMPI that is from software industry. Because of interdisciplinary study, in this paper we focus on reviewing the applicability of the interdisciplinary application and establishing a draft framework. While existing educational appraisals that mainly emphasize organizations, the proposed assessment framework is designed to further link and accredit TQ to teachers. It is hoped that by presenting this preliminary work in this prestigious conference, more constructive and valuable comments and suggestions can be obtained for the future development of the individualized TQ assessment research.*

Keywords: Higher education, teaching quality, teaching capability maturity

1 Introduction

Like any organization that stresses the importance of quality management, schools are also responsible for providing quality teaching. In higher education, teaching quality (TQ) is also a primary focus. To address TQ in higher education, besides existing methods and standards that focus mostly on the organizational level, attention should also put to teachers who executively determine TQ. In addition to the call on teachers, process also plays a critical role in quality teaching (Grant et al., 2004; Louden, 2000). Yet existing evaluations on a teacher's TQ, such as course evaluations or peer review, largely depend on students' subjective feeling or the teacher's momentary performance during the peer visit (*Cramer* and *Alexitch*, 2000; Marsh and Roche, 1997). This is sometimes too judgmental or partial, neglecting other efforts a teacher may have made during the processes of the entire course.

In the area of process-focus teaching quality, recent study has developed a Teaching Capability Maturity Model (T-CMM) (Chen et al., 2014; Chen et al., 2011). T-CMM is designed to address TQ by comprehensively covering the endeavors of a teacher in the lifecycle of instructing a course. Basing on T-CMM, we focuses further on the assessment for determining teaching capability and maturity. This is motivated by the fact that T-CMM provides a gradual way for a teacher to continually improve TQ, yet it requires a means for identifying where (i.e. the level of capability or maturity of teaching) of the teacher on the roadmap to superior quality of teaching. Therefore, this study presents a preliminary assessment framework that is based on T-CMM. The proposed TQ assessment framework is designed for individualized (i.e. teachers) usage of teaching capability/maturity assessment; and it differentiates the organizational scope in traditional appraisals.

The entire research of TQ assessment framework comprises two parts: a conceptual study of the applicability review of the framework, and the development and implementation of the framework. This paper serves as the first part, as well as for acquiring suggestions and comments for the next stage of development. Because the framework applies the concepts of SCAMPI that is from software industry, in this paper we focus on reviewing the applicability and then deriving the proposed design features. The rest of this paper is organized as follows: Section 2 reviews the teachers' teaching capability maturity (T-CMM) model; Section 3 reviews SCAMPI; Section 4 introduces the current development of the proposed TQ assessment framework, including the applicability analysis and the process draft design; Section 5 discusses the framework; Section 6 brings up the upcoming research of the TQ assessment framework.

2 Teaching Capability Maturity Model

The Teaching Capability Maturity Model (T-CMM) (Chen et al., 2014; Chen et al., 2011) was establish based on the need of focusing on teachers and their teaching processes, not just solely the in-class performance. T-CMM is an interdisciplinary application; it applies the concepts of CMMI that is from the software industry to the management of TQ in higher education. From the process aspect, the core activities or so-called building blocks in teaching can be modeled based on the curriculum lifecycle of the course. In T-CMM, these building blocks refer to common teaching process areas

(TPAs). Each TPA has specific goals (SG) for a teacher to achieve in implementing the TPA. Table 1 below displays the collection of TPAs and the associated SGs, which are identified from the survey of literature in educational service and higher education.

Table 1: The TPAs in T-CMM (Chen et al., 2014)

Note: For the references that support the contents of TPAs, see: Chen, C.Y., Chen, P.C., & Chen. P.Y. (2014) Teaching quality in higher education: an introductory review on a process-oriented teaching-quality model, *Total Quality Management & Business Excellence*, 25(1) 36-56

TPA	TML	Specific Goals (SG)
Course and Teaching Planning (CTP)	2	SG1 Identify course scope (Barnett, 1992; Brennan et al, 1997) SG2 Establish and maintain a course plan (Ramsden, 2003, SEI, 2011)
Course Requirement Development (CRD)	2	SG1 Develop course requirements (Barnett, 1992; Popham, 1993) SG2 Maintain the requirements (Miller, 1974)
Course Monitoring & Control (CMC)	2	SG1 Monitor & control the course against the plan (Marzano et al., 2005; Martin et al., 2004; Ramsden, 2003; ACM, 2002; Lawn, 1991) SG2 Resolve issues and manage to closure (SEI, 2011) SG3 Manage classroom atmosphere (Brooks, 1994; Marzano et al., 2005)
Course Contents Management (CM)	2	SG1 Establish course configuration (Lutteroth et al., 2007; Ramsden, 2003; Harvey, 1993)
Learning Verification & Teaching Validation (VAL)	3	SG1 Prepare for evaluation (Prince & Felder, 2007; Bloom et al., 1981; Cramer & Alexitch, 2000) SG2 Conduct the evaluation (Bloom et al., 1981; Dill, 1997; Dressel, 1976; Marsh and Roche, 1997; McKeachie, 1997; Prince and Felder, 2007)
Integrated Course & Teaching Management (ICTM)	3	SG1 Establish defined teaching process (Srikanthan & Dalrymple, 2005; Miller, 1974; Trigwell and Prosser, 1991) SG2 Collaborate with other teaching stakeholders (Chen, 2009; SEI, 2011) SG3 Maintain a shared vision with the organization and collaborative stakeholders (SEI, 2011; Trigwell and Prosser, 1991)
Teaching Process Focus (TPF)	3	SG1 Find out teaching or learning related problems (Daigle et al., 2003; Probst et al., 2002) SG2 Establish teaching assets (SEI, 2011, Harvey, 1993; Seidin, 1997)
Quantitative Course Management (QCM)	4	SG1 Establish quantitative quality goals (Brooks, 1994; SEI, 2011) SG2 Manage teaching/learning performance (Marzano et al., 2005; SEI, 2011)
Quantitative Process Management (QPM)	4	SG1 Establish quantitative process goals (Brooks, 1994; Ramsden, 2003; Wilson, 1998) SG2 Manage process performance by numbers (Ramsden, 2003; Wilson, 1998)
Teaching Innovation (TIA)	5	SG1 Establish the environment for teaching innovation (Brooks, 1994; Costantino and Lorenzo, 2000; SEI, 2011) SG2 Review the results of innovative changes (Ramsden, 2003; SEI, 2011)

Based on the establishment of TPA, T-CMM defines six levels of teaching capability and five levels of teaching maturity. Teaching capability level (TCL) refers to how well a teacher performs and sustains the performance of a particular TPA. Teaching maturity levels (TML) refer to the stages in which corresponding TPAs converge for a teacher to implement, in order to achieve an overall teaching performance. The capability level of a particular TPA is determined by implementing the SGs for the TPA and the GG (the goals to sustain the implementation results) associated with the TCL. The maturity level is determined by implementing a set of the TPA that are constellated in the level and achieving the GG that is associated with the TML. Table 1 also displays the constellation (in terms of TML) of identified TPAs. The collection of teaching process areas represents the systematic and comprehensive perspective regarding a teacher's teaching profile/portfolio. Refer to (Chen et al., 2012) for the detailed specific practices (SP) for the SGs in a TPA.

In aspects of the two level representations, a teacher or the teaching team (i.e. with teaching assistants) may implement a group of TPAs collectively in order to reach a quality plateau. Each stage/plateau (i.e. TML) defines an overall topic (e.g. "basic course management" of TML-2) of teaching quality and is associated with related TPAs for the teacher to implement collectively. Alternatively, a teacher can focus on individual TPAs. Such an implementation effort is relatively smaller but remains continuous till all TPAs are implemented and GGs are achieved. Figure 1 below denotes the levels of teaching capability and maturity.

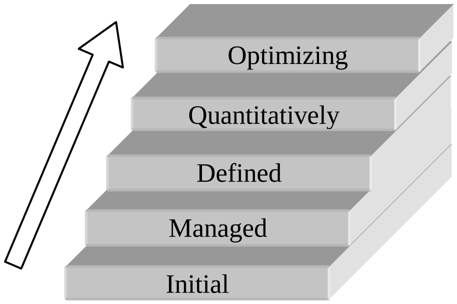
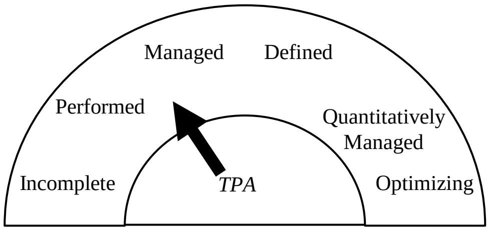
Maturity Level (ML)	Capability Level (CL)
 Optimizing Quantitatively Defined Managed Initial	 Managed Defined Performed Quantitatively Managed Incomplete TPA Optimizing
1. ***Initial*** ： teaching is chaotic in order to complete a course	**0.** ***Incomplete***: cannot satisfy all the specific goals (SG) in a teaching process area (TPA).
2. ***Managed***—*basic course-management*. At this stage, course management related teaching processes areas are managed and controlled. Similar courses would have a similar teaching performance.	**1.** ***Performed***: satisfy all the SGs in a TPA
	2. ***Managed***: the performance of conducting a teaching process area is similar in similar courses
3. ***Defined***— *defining & improving course development*. At this stage, teacher teaching processes are defined and are able to be tailored. Teaching improvement practice begins to establish and sustain.	**3.** ***Defined***: a teaching process area is defined and is able to be tailored. Process improvement is established and standardized for improving the process area.
4. ***Quantitatively Managed***—*quantitative teaching and quality management*. At this stage, teacher quantitatively manages all teaching processes by quantitative objectives.	**4.** ***Quantitatively Managed***: a teaching process area is quantitatively managed based on established quantitative objectives.
5. ***Optimizing***—*continually optimizing Teaching quality*. At this stage all teaching process areas are continually improved and sustained at their optimal situation.	**5.** ***Optimizing***: a teaching process area is continually improved and sustained their optimal situation.

Fig. 1: The teaching maturity and capability levels

3 SCAMPI

SCAMPI, i.e. Standard CMMI appraisal method for process improvement, is designed for software organizations to examine how they have achieved and implemented the process requirements defined in the CMMI for software development (Staples, et al., 2007; Bush, 2005). SCAMPI is a decision support approach with an organizational appraisal scope to help determine the CL/ML level of a software organization and identify its strengths and weaknesses in software development (Ali and Ibrahim, 2011; Wilkie, et al., 2005). In addition to the accomplishment of a CL/ML, organizations regard the findings as the next goal for reaching superior quality of software development.

In SCAMPI, appraisers are provided with actual projects that are conducted by the appraised organization as the evidences of process implementation. Evidences are the artifacts such as the data and documents produced in the appraised projects and relevant to the focused processes, and the interviews with relevant stakeholders to help confirm the execution of the processes. The procedure of SCAMPI, according to SEI (2006), comprises of three stages: planning and preparing for appraisal, conducting appraisal and reporting results. The preparation stage consists of five steps: determine the appraisal scope, develop an appraisal plan, prepare an appraisal team, obtain the evidences and prepare for appraisal conduct, i.e. the readiness review prior to the appraisal. The appraisal stage includes six steps: prepare participants, examine the evidences, document objective evidences, verify the evidence, validate preliminary findings and generate appraisal results. The reporting stage includes 2 parts: deliver appraisal results and package and archive appraisal assets.

In the evaluation of process implementation of specific practices, SCAMPI characterizes it into four levels, i.e., fully implemented (FI), largely implemented (LI), partially implemented (PI), and not implemented (NI) (SEI, 2006, p.97). A specific or generic goal is rated Satisfied if and only if both of the following are true: (1) all associated practices are characterized at the organizational unit level as either LI or FI, and (2) the aggregation of weaknesses associated with the goal does not have a significant negative impact on goal achievement (SEI, 2006, p.158). In the case of capability rating, a CL is determined when the required generic goals are satisfied. For example, a process area has achieved CL-2when the generic goal that characterizes this level is satisfied. In the case of maturity rating, a ML is determined when its aggregated process areas have reached the same level of CL. For example, an organization has achieved ML-2 if all of the process areas constellated in this maturity level have achieved CL-2.

4 The Draft Framework

4.1. Applicability review

In this section, the idea and features of the assessment framework is presented. Because the assessment framework conditionally applies the concepts of SCAMPI, this paper reviews the applicability. The review centers on comparing the two assessment methods in four aspects: appraisal scope, verifying focus, operating environment, and appraisal effort and cost.

The first applicability issue refers to the differences in appraisal scope. Because software development is teamwork with a goal to preserve knowledge in organization, SCAMPI has an organizational appraisal scope and the process is more complicated and involves substantial documentation (Yucalar and Erdogan, 2009; Wu et al., 2006). In higher education, since university teachers are independent and the improvement of TQ is based on individual needs, the assessment framework has an individualized assessment scope. Hence, in applying the process of SCAMPI, the assessment framework requires tailoring to have a simpler process and reduced documentation workload due to its individualized scope.

The second applicability issue refers to the verifying focus. To fully realize how an organization does in meeting the process requirements defined in CMMI, SCAMPI examines the organization's projects, since they provide evidences of comprehensively covering the software development processes. To the TQ assessment in higher education, the assessment framework evaluates a teacher's TQ by examining the courses that are taught by the teacher. This is because that course data actually reflects the teacher's teaching methods and can demonstrate the teacher's actual implementation of the requirements defined in T-CMM.

The third issue pertains to the operating environment of appraisal. To SCAMPI, the operation is centralized; it is often carried out on location of the assessed organization for the accessibility of people and data. Such a centralized operation is reasonable, since SCAMPI has an organizational scope and most of the participants are from the assessed organization. To the assessment framework, because of the teaching independence, course data are often controlled and maintained by teachers, making the TQ assessments less possible to be centralized, i.e. collecting teachers for the assessment. The assessment should be design to resolve this particular situation.

The fourth applicability issue is appraisal effort and cost. Due to the aforementioned organizational scale and assessment scope, SCAMPI is conducted in team and the capability/ maturity rating is a group decision. SCAMPI also often involves considerable costs, due to the scale of the assessment scope, the requirement of the operating environment and the scale of appraisal team. To the assessment framework that has an individualized scope, it may not require a team to conduct a teacher's TQ assessment, as well as for reducing the assessment cost. Nevertheless, the rating may be arbitrary or subjective if the assessment has less or no team support.

4.2. Process framework

Fig.1 is the process framework of the assessment framework. As the figure shows, a TQ assessment comprises two stages: preparation (P) and assessment & reporting (AR). Because the assessment service is designed to be requested on demand, the assessment framework is instantiated and enters the preparation stage when a teacher submits an assessment application. During the initialization, the teacher is required to plan the coverage of the assessment in terms of a particular TML or a particular TCL on certain TPA(s). Because T-CMM imposes a gradual and contextual approach of TQ improvement, the coverage is based on prior assessment results. Therefore, as Fig.2 shows, the assessment framework provides previous assessment (if any) information, for the current assessment to be planned on a contextual and accumulative basis.

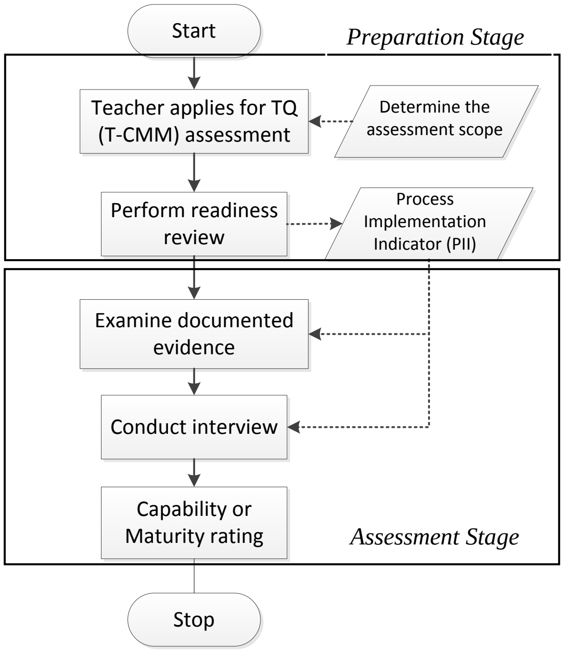

Start
Preparation Stage
Teacher applies for TQ (T-CMM) assessment
Determine the assessment scope
Perform readiness review
Process Implementation Indicator (PII)
Examine documented evidence
Conduct interview
Capability or Maturity rating
Assessment Stage
Stop

Fig.2: The process framework of the TQ assessment

Also, as the figure illustrates, P consists of a step that collects the data for this assessment based on the coverage. By basing on T-CMM, the data to be collected refer to the evidences that are provided by the teacher to show his/her implementation towards the specific and generic practices (SP/GP) for the to-be-assessed TPAs (see Table 1). Besides, the assessment framework rates TCL or TML based on the teacher's real course cases, hence the evidences refer to the actual data and documents produced in the course. The data also include the name list of course participants, i.e. students, teaching assistants, invited speakers (if any), for the purpose of selecting and inviting them to the interview in the AR stage. Moreover, the assessment framework provides a Process Implementation Indicator (PII), a spreadsheet (e.g. MS EXCEL) for the teacher to upload the evidences and files. As illustrated in Fig.2, the PII provides a summary view for the teacher to check for readiness before entering the AR stage, as well as for the appraiser to access the evidences easily during the assessment.

After the preparation is complete, the assessment framework enters the AR stage to perform the assessment. In this stage, an appraiser is designated for the assessment. The selection of appraisers may affect the quality of the assessment, thus it will be further addressed later in discussing the practical issues of the proposed work. The technical part in this stage refers to the TCL/TML rating, which consists of (1) the evaluation of degree of process implementation against the SGs/SPs and GGs/GPs of TPAs defined in T-CMM, and (2) summarizing the overall findings to obtain the resulting TCL/TML level. Once TCL/TML is obtained, the assessment framework performs peer review to ensure the quality of the appraisal results.

5 Discussion

In this section, the process design of the assessment framework is discussed. The discussion is expected to gather valuable comments and opinions via this conference venue, for the future development and usage of the assessment framework.

- *The tailoring of the SCAMPI process*

In spite of the different in assessment scope, both SCAMPI and the assessment framework have the same focus on, that is, process examination. With the same emphasis, the assessment framework applies and tailors SCAMPI for the educational domain and individualized use of TQ assessments. In reviewing such a tailoring, this study follows the leveled approach, i.e. lifecycle, procedure, role, and work-products (Chen and Huang, 2009; Fitzgerald et al., 2003). First, for the tailoring of assessment lifecycle, the number of stages in SCAMPI is reduced to 2 in the assessment framework. Second, for the process level of tailoring, the steps P1, P2, P3 are integrated into one step, i.e. applying TQ assessment, in the assessment framework; the steps. For the role tailoring, the proposed work preserves the appraiser and role, thus the procedures performed by the appraiser are preserved in the proposed process design. For the work-product level, the proposed work simplifies the artifacts of appraisal plans, appraisal reports and PIIs by incorporating them into the system for an online production of the work products to support the reduced scale and scope of TQ assessments.

- *Assessment on-demand and reducing the assessment workload*

Owing to the characteristics of individualization and dispersedness of participants, the assessment should be requested on-demand. That is, a teacher should be able to individually request for an assessment to certify different TML/TCL levels. Besides, care must be taken in reducing the workload for such an individualized assessment. For a TQ assessment to be able to request on demand, the assessment framework is expected to be associated with Information System and Quality (ISQ) Laboratory that would launch an experimental administrative program, including applying for TML-TCL assessment, recruitment of appraisers and appraisal data maintenance. The assessments would be available shortly when the experiment is set. However, due to a research prototype, the assessment is expected to be available only in Taiwan area.

To reduce the workload in preparing and performing such an assessment, information technologies may be considered as a solution. In this regard, the proposed framework and the process design will be further implemented into an online assessment system in the next stage of the research. The computerized functions are supposed to facilitate the tedious execution of the appraisal service and to help manage effectively the tremendous volume of assessment data. The computerized implementation should be able to offer the convenience for a teacher to upload the evidences, as well as for the appraiser to view the contents by clicking on the links. Moreover, online messaging or live talk functions should be provided with participants for operating a more flexible interview in such a TQ assessment.

6 What is Next ?

In schools, teaching quality has been playing a major role in the success of education. As existing research of educational quality mostly focuses on organizational benefits, the proposed work in this paper is designed to accredit teaching individuals and teachers. The design of the teaching capability and maturity assessment provides an individualized and quantifiable roadmap to help teachers improve teaching quality.

In the next stage of research, we are to further cooperate with an educational research laboratory (i.e. ISQ) for developing the assessment framework. In addition, we are to develop an online assessment system, for the assessments to be more flexible and effective. Third, we are to apply some information technologies, to help address the subjectivity of manual justification when determining the levels of process implementation in TQ assessments. The development of the work and the promotion to the society has been with an ultimate shared goal: towards superior education quality.

Acknowledgement

We would like to thank Taiwan’s National Science Council for granting and financially supporting this research (2013-2014)[No: 102-2221-E-008 -054].

References

[1] ACM, AIS, AIPT (2002). *IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*, Association for Information Systems

[2] Barnett, R. (1992). *Improving Higher Education*, SRHE and Open University Press: PA.

[3] Brennan, J., edVries, P. and Williams, R. (1997). *Standards and Quality in Higher Education*, Jessica Kingsley Publisher: PA.

[4] Brooks, K. (1994). *Total quality Teaching: Microanalysis of Effective Teaching Practice*, Texas: Dallas.

[5] Chen, C.Y., Chen, P.C., Chen. P.Y. (2014) Teaching quality in higher education: an introductory review on a process-oriented teaching-quality model, *Total Quality Management & Business Excellence*, 25(1) 36-56.

[6] Chen, C.Y., Kuo, C.Y., Chen, P.C. (2011). The teaching capability maturity model for teachers in higher education: a preliminary study, *2011 International Conference on Frontiers in Education: Computer Science and Computer Engineering*.

[7] Costantino, P., Lorenzo M. (2002). *Developing Professional Teaching Portfolio*, Pearson Education Company, MA.

[8] *Cramer,* K.M., *Alexitch*, L.R. (2000). Student evaluations of college professors: identifying sources of bias. *Canadian Journal of Higher Education*, 30(2), 143-164.

[9] Daigle, R.J., Longenecker, Jr. H.E., Landry, J.P., Pardue, J.H. (2003). Using the IS 2002 model curriculum for mapping an IS curriculum. *Proceedings of ISECON 2003.*

[10] Dill, D.D. (1997). Accreditation, assessment, anarchy? The evolution of academic quality assurance policies in the United States, Ed by Brennan, J. *et al*. *Higher Education Policy Series 37.*

[11] Dressel, P.L. (1976). *Handbook of Academic Evaluation*, San Francisco: Jossey-Bass.

[12] Fitzgerald, B., Russo, N., O'Kane, T. (2003), Software development method tailoring at Motorola, *Communications of the ACM*, 46(4), 65-70.

[13] Grant, D., Mergen, E., Stanley. W., (2004). A comparative analysis of quality management in US and international universities. *Total Quality Management,* 15(4), 423–438.

[14] Harvey, L. (1993). Continuous quality improvement: a system-wide view of quality in higher education, in Knight, P. (Eds). *System-wide Curriculum Change*, SEDS, Oxford University Press, Oxford,

[15] Lawn, M. (1991). Social construction of quality in teaching. *Evaluation and Research in Education*, l5(1).

[16] Louden,W. (2000). Standards for standards: the development of Australian professional standards for teaching. *Australian Journal of Education*, 44(2), 118-134.

[17] Lutteroth, C., Luxton-Reilly, A., Dobbie G., Hamer, J. (2007). A maturity model for computing education. *Proceedings of the ninth Australasian conference on Computing education*, 66, 107-114.

[18] Marsh, H.W., Roche, L.A. (1997). Making students' evaluations of teaching effectiveness effective: The critical issues of validity, bias, and utility. *American Psychologist*, 52, 1187-1197.

[19] Martin, V.A., Hatzakis, T., Lycett, M., Macredie, R. (2004). Building the business/IT relationship through knowledge management. *Information Technology Cases and Applications*, 16(2), 27-47.

[20] Marzano, R.J. Gaddy, B.B., Fseid, M.C., Foseid, M.P., Marzano, J.S. (2005). *A handbook for classroom management*, Association for Supervision and Curriculum Development: VA.

[21] McKeachie, W.J. (1997). Student ratings: The validity of use. *American Psychologist*, 52, 1218-1225.

[22] Miller, R.J. (1974). *Developing programs for faculty evaluation*, San Francisco: Jossey-Bass.

[23] Prince, M., Felder, R. (2007). The many faces of inductive teaching and learning. *Journal of College Science Teaching*, 36(5), 14–20.

[24] Popham, W.J. (1993). *Educational Evaluation*, Simon and Schuster Publishing Inc.: MA

[25] Probst, G., Raub, S., Romhardt, K. (2002). *Managing Knowledge: Building Blocks for Success*, New York: John Wiley & Sons, Ltd.

[26] Ramsden, P. (2003). *Learning to Teach in Higher Education*, Taylor and Francis Group: NY.

[27] SEI (2011). *CMMI for Development v1.3*, Carnegie Mellon University Press, PA: Pittsburgh.

[28] SEI (2006). *The SCAMPI Method for CMMI evaluation*, Carnegie Mellon University Press, PA: Pittsburgh.

[29] Seidin, P. (1997). *The Teaching Portfolio in Higher Education*, Anker Publishing Company, Inc.: MA.

[30] Srikanthan, G., Dalrymple, J. **(2005).** Implementation of a holistic model for quality in higher education, Quality in Higher Education, 11(1), 69-81.

[31] Williams, P. (2008). A practical application of CMM to medical security capability. *Information Management & Computer Security*, 16(1), 58-73.

[32] Wilson, R. (1998). Report blasts research universities for poor teaching of undergraduates. *Chronicle of Higher Education*, 44(33), A12-A13.

[33] Wu, Z., Christensen, D., Li, M., Wang, Q. (2006). A Survey of CMM/CMMI Implementation in China, *Lecture Notes in Computer Science*, Volume 3840, 507-520.

[34] Yucalar, F., Erdogan, S.Z. (2009) A questionnaire based method for CMMI level 2 maturity assessment. *Journal of Aeronautics and Space Technologies, 4(2), 39-46*

Institutional Effectiveness Assessment Collaboration Study Based Critical Thinking Approach

Ghassan Azar
Lawrence Technological University
Southfield, MI, USA
gazar@ltu.edu

Su Yu
Shanghai University of Engineering and Science
Shanghai,China
suyu_sh@hotmail.com

Clay Gloster
North Carolina A&T State University
Greensboro, NC, USA
cgloster@ncat.edu

Naser El-Bathy
North Carolina A&T State University
Greensboro, NC, USA
nielbath@ncat.edu

Xincheng Xu
Shanghai University of Engineering and Science
Shanghai,China
xincheng@sues.edu.cn

***Abstract*—The academic education system confronts major challenges to graduate high quality students. Therefore, schools started to implement assessment tools to measure the level of the students, faculty, and teaching methodologies in America. However, the current education system in China is sufficient to graduate students with knowledge in their program of studies. The equipment and technology provided for students are superior. Large number of students in classes are challenged and tested to pass their exams. The current system emphasizes on memorization of the subject matter. Little or no attention is devoted to enhance students' critical thinking. In this paper, a novel institutional effectiveness assessment collaboration study based critical thinking approach is presented. This study is a result of collaboration educational plan developed by three schools in America and China.**

Keywords-Critical Thinking; Effectiveness; Assessment; Collaboration

I. INTRODUCTION

Development of critical thinking skills and activities has become a major criteria for successful education programs. In computer science and information technology programs, students proved better capabilities in performing critical thinking activities [1]. As a result, many schools design new curriculum to expand technology learning throughout its programs. However, implementation of appropriate new curriculum requires solution for current educational dilemmas. Theoretical professors face typical challenges. These include large number of students in classrooms, poor motivation techniques, heavy load of course study materials, and class time-wasting [2].

A solution is based on time administration, students learning motivation, plan of study, and exam preparation skills, classroom interactive engagement, technology teaching involvement, and execution of cooperative and problem-solving methodologies. The solution must encourage students to take advantage of instructors and teaching assistants office hours, reduce cheating, and support students during their emergencies' situations [2].

This paper introduces an institutional effectiveness assessment and improvement collaboration study that maintains critical thinking approaches to achieve the goals of students in 3 schools in America and China. The study defines 5 procedures to improve instructors' lecture grounding, students motivation, overloaded syllabi, educational technology tools and methods, supportive learning, homework grading policy, and test design. Figure 1 depicts these five procedures.

The rest of this paper is structured as follow: Section two identifies the success factors and procedures. Section three presents the Chinese critical thing, Section 4 introduces the assessment study and finally the conclusion is given.

II. SUCCESS FACTORS AND PROCEDURES

Figure 2 illustrates the priority of each of the five procedures. Lecture preparation and supportive learning have higher priority. Then student motivation and technology tools have less priority. The first procedure requires that the instructor uses 9 hours to prepare a lecture for a new 3 credit hour course. This includes reviewing the lecture materials, assignment, in-class activity, and asking and answering questions. Instructors should avoid a common mistake that is spending extra hours on lecture preparation. New faculty members always make such mistake [2].

The second procedure emphasizing on motivating students utilizing more advanced techniques such as calling the students by their names, demonstrating the objectives of measurable learning, and introducing education success factors. Motivation can also be achieved via allowing assignment/exam second chance.

The third procedure enhances students' creativity and skills by allowing them use new technology tools not only in laboratory but also in class. The fourth procedure maximizes the supportive learning by encouraging team work to solve theoretical and technical problems. Group exam is an intelligent methodology that improves students’ critical thinking abilities. The fifth procedure is based on a combination of overloaded syllabi and test design. The syllabus should clearly explains the course objective goals and learning outcomes. The test should be based on lecture and textbook and supplemental materials reading.

III. CHINESE CRITICAL THINKING

In China, the current education system in general and at Shanghai University of Engineering and Science (SUES) specifically is sufficient to graduate students with knowledge in their program of studies. In fact, the equipment and technology provided for the students meet and/or exceed what is provided at most universities in the United States [3]. Students in China are challenged and tested on their abilities to memorize different subject matters and be able to pass their exams. The ability to memorize learned material is important especially in Mathematics.

With the development of Engineering Training Center, new ideas and training platform for mastering the conventional technical skill are created. The model of typical cases actuation to develop innovative practice education is introduced. Teaching achievements about bringing up student’s creative consciousness and innovative ability are concluded.

Here, the course “Modern Manufacturing Technology Practice ” is an example. This course features quality education and capability with hand, which stresses not only the teaching of basic theory but the training of students’ practice competence. How to integrate with each other in content is very important.

This course includes several ideas, modern manufacturing technology basis, different kind of machinery, part material, structural design, technology design, manufacturing process simulation, assembly, and accuracy. In traditional teaching, the students will be in different practice units to learn this knowledge. Each unit is independent. The questions are how to let students have overall concept of modern manufacturing technology and how to let students have the ability to solve questions between units.

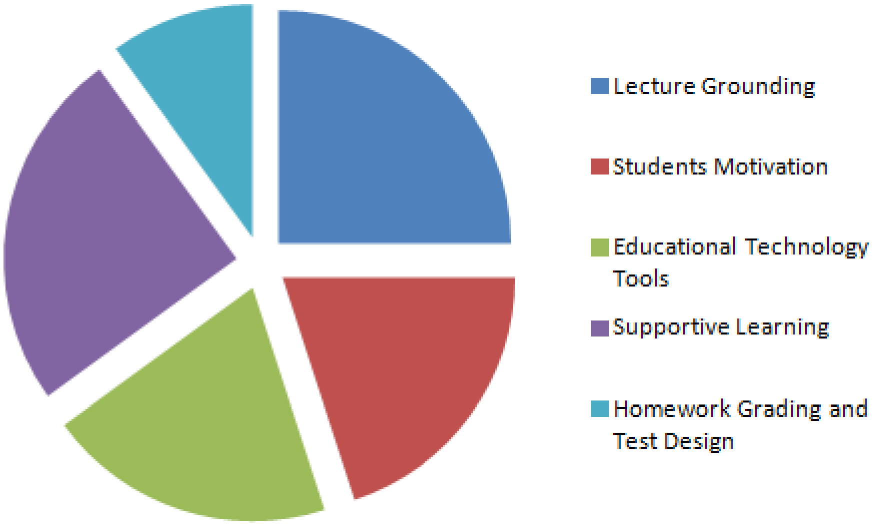
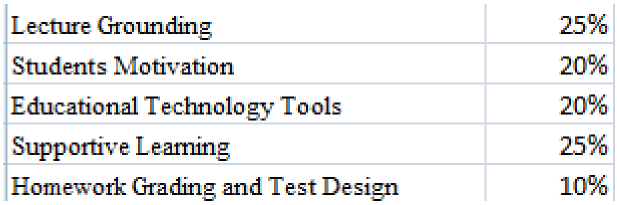


Fig. 1 The study 5 procedures



Lecture Grounding	25%
Students Motivation	20%
Educational Technology Tools	20%
Supportive Learning	25%
Homework Grading and Test Design	10%

Fig. 2 Procedures percentage

Based on typical cases actuation, new education thought and teaching idea are used in the curriculum teaching. By means of contents' diffusion and combination, students can think by themselves and then to analyze, solve, or comprehend problems. At the hardware and software aspects, the programming and operating of the CNC Lathe and NC Miller, Master CAM or CAD/CAM software, EDM Machining, NC caver, three dimensional measuring machine, rapid prototyping machine are used in this course. At the education aspect, typical cases are designed, for example, crisscross groove cohesive shift coupler transmission system, universal coupling cohesive shift coupler transmission system and so on. These case actuations combined with different separately job that be in finished in different practice units, such as different jobs, different machines, dimension measurement, parts manufacture, assemble and so on. In case crisscross groove cohesive shift coupler transmission system includes practices as follow: CAD design, manufacturing machine selected, part structure technological analysis, technological process design, CAM programming and simulating, part manufacture, dimension inspection, parts assemble, summarize. If one of them is wrong, the student can't do others well.

A typical class on an average consists of fifty to seventy students. This is too large of a classroom to educate students properly and especially for scientific subjects. A Faculty member lectures for an hour or more using a microphone while following some Microsoft PowerPoint slides. These slides are generated by the faculty member specifically for the class [3].

In general, Chinese text books are difficult to digest and special attention needs to be given to more efficient text books. These text books must be devoted to the subject area and provide resources to the faculty and the students [4]. We observed that some students in the class listening to the lecture. Other students might be doing other tasks like homework for other classes and not paying attention to the lecture at hand.

In the United States, we break our lectures in five to seven minutes segments. At each segment we strive to engage students by either asking questions or sharing examples of what was lectured in the last segment. Students' involvement is the key to getting students engage in critical thinking and/or analysis.

Critical thinking and ability to analyze new situations will lead to innovation and creative thinking. Innovative and creative thinking will put these students at a clear advantage in today's competitive economic world. We need to graduate students that can think on their feet and hit the work force running and possess innovative edge. They must be able to dynamically solve new problems; which have not been encountered during their academic studies [4].

Problem Based Learning (PBL) is one of the latest methods utilized by our university to engage students honing their ability to solve problems. PBL provides students with the setting to exercise critical thinking in solving problems while gaining new theoretical knowledge [5]. Applying PBL requires student to read the chapters ahead of time and be prepared during the lecture to be engaged discussing and solving problems. Hence the teacher is not delivering the theory alone, but encompassing the practice as part of the lecture. Therefore, the teacher must work harder as well by preparing several sample problems to embed within each lecture. Students should be encouraged to develop new ways to solve problems; hence we achieve critical thinking [5].

PBL provides foundation for students to be able to think independently and not programmed to follow some pre-given instructions like a robot. Professor Yang Dongping, a researcher at the Beijing Institute of Technology, warned that students in the Chinese system are not being developed with critical thinking [6].

Active Collaborative Learning (ACL) is another method utilized by Lawrence Technological University (LTU) to build students' ability to work in teams [7]. After a segment of theoretical lecture we break our students in small teams containing three to five students to discuss the material lectured and shed a light on their understanding and possible enhancement to the theory at hand. Students must be challenged to brainstorm in a group to simulate work environment. Some students might copy existing superior ideas and argue that doing so is equivalent to inventing and innovating. All presented ideas should be discussed and reward(s) should be given to original idea(s). At SUES, teams are used in competitions or when a set of students are interested in an area of study such as game development [7]. The ability to work in a team or applying ACL is not enforced in regular classes. Therefore, not all students at SUES experience working in teams. Team work is essential to individual and group success at any organization [7].

IV. ASSESMENT STUDY

In the School of Technology at North Carolina Agricultural and Technical State University (NCAT), the department of Computer Systems Technology (CST) follows common program outcomes and student learning outcomes for the Master of Science in Information Technology educational program. The three student learning outcomes for the MS in Information Technology program are summarized in Table 1 showing the relationship between outcomes and the assessment and results and the improvements made.

Students in the MSIT program will be able to communicate information technology system functionality with the clarity and precision required by standards commonly practiced in the profession. This outcome was assessed in ITT650: Wireless Communication Systems I. The measure used for this outcome was homework problems that required students to complete learning summaries demonstrating their abilities to understand foundational concepts of wireless communication systems and communicate them effectively while addressing the required specifications. The data indicated that students in ITT 650 were highly proficient in summarizing class content and communicating the information effectively. During the Spring 2012 semester, 75% of the students scored more than 80% on the assessments. During the Fall 2013 semester, 87.5% of the students scored 80% or more on the assessments.

Students in the MSIT program will be able to communicate information technology system functionality with the clarity and precision required by standards commonly practiced in the profession. This outcome was measured in ITT 700 (Project Management for IT Professionals) from 2008 through 2012, and continues in 2013.

- A team-based scenario synthesis project was introduced in 2010, with a video DVD presentation required of each team. This project cast students in the role of Project Managers tasked with the specification, installation, commissioning, and acceptance testing of a large high performance Beowulf Cluster modeled on an actual installation at our sister university, UNC-G.
- In 2011, a Blackboard-based collaborative wiki was added to allow teams interactive collaboration both in small teams and as a member of the full class team to emulate a national corporate environment.
- In 2012, the wiki was implemented as a standalone Internet wiki to emulate a global corporation, and multiple, written, audio, and video deliverables added.

Table 1. Student Learning Outcomes, Assessments, and Improvements

Name of Program	Student Learning Outcome	Method of Assessment	Results of Assessment	Use of Assessment Results for Improvement
Master of Science Degree in Information Technology (MSIT)	Students in the MSIT program will be able to communicate information technology system functionality with the clarity and precision required by standards commonly practiced in the profession. Students in the MSIT program will be able to diagnose and resolve technical problems in information technology systems.	This outcome was measured in ITT 650 (Wireless communication Systems I) and ITT 685 (Ethical Issues in Information Technology). The homework problems used for ITT 650 required students to complete learning summaries demonstrating their understanding of concepts of wireless systems. The assignments for ITT 685 were written assignments. For ITT 650, the outcome was measured through homework and exam questions targeting the student's ability to determine the number of RF channels required and the number of permissible calls per cell for a CDMA wireless network	For ITT 650, during the Spring 2012 semester, 75% of the students scored more than 80% on the assessments. The number of students increased by 12.5% in the Fall 2012 semester when 87.5% of the students scored 80% or more on the assessments. The outcome was also assessed in ITT 685. During the Spring 2012 semester, 92% of the students in the class achieved the target level on written assignments in which students had to articulate a position on ethical issues concerning, privacy, intellectual property, and civic responsibility associated with technology usage.	The graduate exit survey was revised during the 2011-12 academic year to better align with the program outcomes, particularly to gather specific date related to communication. Particular questions on the examination not only gather data on student job placement, but also require the student to perform a self-evaluation with respect to specific outcomes, i.e. communication. In an attempt to increase the pass rate of students on the comprehensive examination and to ensure the student's competency in information technology and their ability to recommend effective and efficient IT systems, the department faculty convened and made the following improvements: 1) Provide additional sample problems from past examinations for students to use as practice problems during study. 2) Ensure the availability of faculty during office hours. These strategies were implemented during the 2012-13 academic year.

Also, in 2011, interactive two-way video web conferences were added to further enhance the emulation of a global enterprise environment. Students held ad hoc team web conferences, as well as leading class lecture and discussion on a scheduled basis.

Specific outcome wording for the course outcomes in the syllabus evolved from nothing in 2008 to the inclusion of specific graduate program outcomes in 2013. Similarly, outcome assessment and evidential archiving evolved from an ad hoc process to a formal, department-wide process in 2013. In each of the four years, more than 90% of students scored above the 80th percentile on this assignment, as should be expected among graduate students. However, the consistent scoring level was maintained on an increasingly complex, technology-rich project, indicating a measured, consistent growth in achievement of the outcome.

Technologists manage and apply technology; engineers design technology. Our graduate program is focused on educating students to prepare them as mid-level IT managers, with a career path to executive corporate management. ITT 700 is an example of a core required course implementing significant revision and enhancement annually to keep pace with the changes in enterprise communication. Ours is an on-line program; many of our students are full-time IT professionals, and are understandably intolerant of irelevant or disorganized coursework. The past four years have seen the same basic outcome morph from an academic communication to a mobile, global, multi-media collaborative dialog, and we have managed the change to realize consistent performance, rather than a disruptive experience. The quantitative measurement of outcome is supplemented by subjective measurement in the form of consistently high student satisfaction in surveys and unsolicited student testimonials.

Students in the MSIT program will be able to diagnose and resolve technical problems in information technology systems. As presented in Table 2, this outcome was assessed in two courses: one 600 level course and one 700 level course. In ITT 650: Wireless Communication Systems I, the measure used was homework and examination questions demonstrating the student's ability to design a 4-cell and 9-cell cellular cluster for a wireless cell phone network. Students were to determine the number of RF channels required and the number of permissible calls per cell for a CDMA wireless network. For ITT 702, students were required to complete a problem on the final examination that demonstrated their ability to use linear regression analysis to reach a conclusion from a dataset related to the problems reported and their technical constraints.

Table 2. Performance in Problem Solving Skills

Problem Solving Skills	Measure	Spring 2011 Result (% on target)	Fall 2011 Result (% on target)	Spring 2012 Result (% on target)
ITT 650: Wireless Communication Systems I	Homework and examination problems	75.0	87.5	
ITT 702: Statistical Methods for Information Technology	Final examination problem	70.0		77.0
Average		72.5	87.5	77.0

The average level of achievement of this outcome for all courses was 79.0%. Hence, students assessed for this outcome performed at an acceptable level.

This outcome was also assessed using the results of the comprehensive examination. The comprehensive examination demonstrates the students' ability to think critically and to recommend the most effective systems based on the rigorous exercise of calculating key system parameters that determine the feasibility of implementing a particular system. On this examination students are forced to rely on their technical knowledge acquired throughout their coursework. Problems completed on the examination require deep thought and test the student's ability to apply the knowledge obtained in these classes to real-world problems. Students enrolled in the coursework option complete a total of two comprehensive examinations, one comprehensive examination for the foundational courses and another for the technical courses.

The courses for the comprehensive examination in the MS in Information Technology degree program are: ITT 787: Master's Foundation Courses Comprehensive Examination and ITT 788: Master's Management and Technical Courses Comprehensive Examination. Students must enroll in ITT 787 for the foundational course comprehensive examination and ITT 788 for the technical course comprehensive examination.

V. Conclusion

The institutional effectiveness assessment collaboration study based critical thinking approach improved the level of students' creativity and ability to learn better and faster. It introduced new measure tools to verify the results of the assessment methodologies.

Large size of classes hinders faculties' abilities to introduce new educational methods; which will improve students' ability to cope in the work force. Both PBL and ACL require students' engagement to have meaningful experiences in critical thinking. Building students' ability to interact and work in team environments is a key to the organization, group, and their own success at work. In addition these methods require more work for both faculty and students for the different way of approaching the delivery of education.

References

[1] P. Wankat, "The Effective, Efficient Professor: Teaching, Scholarship, and Service," Boston, Allyn & Bacon, 2002.

[2] G. McMahon, Critical Thinking and ICT Integration in a Western Australian Secondary School. Educational Technology & Society, 12 (4), 269–281, 2009.

[3] A. Adeyemo, “The buffering effect of emotional intelligence on the adjustment of secondary school students in transition,” Electronic Journal of Research in Educational Psychology, 3 (6), 79-90, 2005.

[4] T. Lee, “Learning with ICT: The challenge of changing the way we teach,” International Conference on Computers in Education, December 4-6, Auckland, New Zealand, 2002.

[5] V. Koller, S. Harvey, and M. Magnotta “Technology-Based Learning Strategies,” SPR, Oakland, CA, 2005.

[6] J. Karim, and B. Ramadan, “Impact Assessment of Problem-Based Learning in an Engineering Science Course,” Journal of STEM Education Volume 9, Issue 3&4 July-December 2008.

[7] J. Buck, and K. Wage, “Active and Cooperative Learning in Signal Processing Courses,” IEEE Signal Processing Magazine, vol. 22(2), pp. 76-81, 2005

Verification and Validation of a Database Management Course

Nasser Tadayon, Connie Nyman

Department of Computer Science and Information Systems,
Southern Utah University, Cedar City, UT, USA

Abstract *- The concept of verification and validation (V&V) is one of the most essential part of any system to ensure that the products and processes conform to the requirements of the system. It would also ensure that the right problem is being solved in order to meet intended use and user needs. Within the required activities for verification and validation, the components of a system are checked to ensure that it satisfies the standards and conventions. In educational field, there are products, services, which as whole constitute a system that continuously goes through verification and validation at all levels by experts in the field. At each level of educational system, there are committees and processes that assess and evaluate (verify and validate) the outcome of the system. The V&V concept is used for checking a course as a component of a program within the higher educational system. The students go through different assessment process to ensure that they meet the requirement of the course and fulfills its intended purpose.*

This paper investigate and explore the verification and validation within educational system as well as how it is established in a specific course, meeting the university level requirements. The paper also contains analysis of specific outcome through V&V process for the course. The course surveyed is a Database Management class offered through department of Computer Science and Information Systems at Southern Utah University.

Keywords: verification, Validation, Assessment, Project, Higher Education

Introduction

A system in general is comprised of interdependent entities, processes, and data stores which interact with each other through dataflow. The educational system in United States is a complex system that has many components like the higher education with sub-components like programs offered within an institute. Although the educational system in US are mainly controlled by the states, at federal level, the department of education has an important role in encouraging education at the higher level. The role of federal government is mainly to provide financial assistance to students and states or funding specific programs that are essential through grant and loans or subsidies.

The department of education has indicated their mission is “to promote student achievement and preparation for global competitiveness by fostering educational excellence and ensuring equal access” [5]. It engages in four major types of activities:

1. Establishes policies related to federal education funding, administers distribution of funds and monitors their use.
2. Collects data and oversees research on America's schools.
3. Identifies major issues in education and focuses national attention on them.
4. Enforces federal laws prohibiting discrimination in programs that receive federal funds.

The states on the other hand, through the state office of education, monitor and fund public educational system at all level by identifying the skills and competencies requirements. At each level, the educational system goes through the process of verification and validation and is compared against other existing systems based on the quality and quantity.

Verification and validation

Verification and Validation are defined as “independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its intended purpose” [1]. There has been many research and publications in the process of verification and validation within variety of systems in simulation or medical [3] but not much within the educational system. There seem to be many departments and committees within the educational system which through annual assessment, re-evaluate and identify the goals and objectives of the educational system at federal and state level based on existing data and the future demand.

In general, there are critical components and standards for a quality management system such as ISO 9000 [4] that are widely used to verify and validate a system. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

It is sometimes said that validation can be expressed by the query "*Are you building the right thing*?" and verification by "*Are you building it right?*". In practice, the usage of these terms varies. Sometimes they are even used interchangeably.

The PMBOK (Project Management Body of Knowledge) guide, a standard adopted by IEEE, defines these terms as follows [2]:

- "Verification: The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation."
- "Validation: The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification."

In order to understand the verification process used at the state level for a higher educational system, we must first define the requirement, the process, and the product. Although the requirement of obtaining a higher educational degree is mostly defined by the degree program but there are specific skills that must be met by all graduates of a higher educational institute. These skills are mostly outlined at some state universities or colleges as the general graduation requirement.

Essential Learning Outcome

The general graduate requirement at higher educational institutes are identified as specific Essential Learning Outcomes (ELOs). Southern Utah University has adapted ELOs using the template provided by the Association of American Colleges and Universities (AACU) [6]. These ELOs are assessed to verify and validate the programs offered by the institute.

These competencies are embedded within the curriculum of all the degree programs. In general, there are accreditation entities for specific program like business or computer science which clearly define the minimal required competencies for graduates within those programs. On the other hand, the product of the higher educational system are indeed the graduate students of the degree programs. The process is well defined within each institution and their programs which are clearly outlined through their catalog. The question of verification of a degree program within a higher educational institute relates to providing evidence that the product conform to the requirements. The validation would correspond to correctness, completeness, consistency, and accuracy of the degree requirement as well as satisfying standards, practices, and conventions during the process.

Although the database management course does not cover all ELOs for an undergraduate program, it emphasizes on some of them. A summary of some of the Essential Learning Outcomes (ELOs) at Southern Utah University related to the database management course are as follows:

INTELLECTUAL AND PRACTICAL SKILLS

2. Inquiry & Analysis

Inquiry: Does the program of study/course require students to systematically explore issues, objects or works through the collection and analysis of evidence that results in informed conclusions or judgments?

Analysis: Does it also require students to break complex topics or issues into parts to gain a better understanding of them?

3. Critical Thinking - Does the program of study/course require students to comprehensively explore issues, ideas, artifacts, and events before accepting or formulating an opinion or conclusion?

4. Creative Thinking - Does the program of study/course require students to combine or synthesize existing ideas, images, or expertise in original ways, as well as think, react, and work in an imaginative way.

5. Communication - Does the program of study/course require students to develop and express ideas; in writing, by speaking, visually, kinesthetically, through design, or aurally?

8. Teamwork - Does the program of study/course require students to productively interact with others (in or out of class) to complete assignments, tasks or projects?

9. Problem solving - Does the program of study/course require students to design, evaluate and implement strategies to answer open-ended questions or achieve desired goals?

PERSONAL & SOCIAL RESPONSIBILITY

13. Lifelong Learning – Does the program of study/course require students to engage in purposeful learning activities, undertaken on an ongoing basis with the aim of improving knowledge, skills, and competence which is directed toward developing learning self-sufficiency?

Course evaluation

Database Management is a required course for several programs. This course is being analyzed to identify V&V for some of the ELOs through the course project. Two external team projects were used during this course to provide students with real world experience. A member of department's IAB (Industrial Advisory Board) who is the cofounder of an application development company was one the customers for the project. The other customer was a graduated student who was working on developing an application with embedded database. The students were expected to gather relevant information needed for developing the database from the customers and design and develop a database to meet their requirements. The following list of ELOs related to the project were assessed for verification and validation:

2. Inquiry & Analysis
5. Communication
8. Teamwork

Verification and validation of the Essential Learning Outcomes

The following is a brief list of some of the activities related to verification and validation of three ELOs mentioned above:

Inquiry & Analysis:

Validation (Are you building the right thing? - external): The students were given a general description of the requirement and went through the process of fact findings. The customer gave a presentation and there were open Q&A sessions with the customer. A requirement specification document was developed by all teams. All requirement documents (SRS) were sent to the customer for possible changes before their final approval.

Verification (Are you building it right? - internal): The SRS document was checked for the format correctness using the IEEE template.

Communication:

Validation (Are you building the right thing?- external): The teams interviewed the customers and were encouraged to obtain additional information from the website and through direct contact with the customers (validate).

Verification (Are you building it right? - Internal): The teams were asked to develop requirement and design documents and present the implementation of the project with running several test cases to the customers (verify).

Teamwork:

Validation (Are you building the right thing?- external): The teams had the option of changing roles as they would deem appropriate. They were also asked to find common meeting times during launch activity. The team had regular meeting off the class time and were asked to keep track of the meeting minutes (validate).

Verification (Are you building it right? - Internal): The teams were set based on the student's transcript, schedule, and preference. They were provided with a set of team roles and were suggested a specific role for each member (verify).

Data analysis

In order to assess the team performance, everyone in the class including the team members and the customers were asked to rank the team based on their individual contributions and presentation. The overall average was calculated using formula: AVERAGE (all evaluation)*30-200)/10. This would eliminate several high scores (noise in data) recorded by some students. There were 20 students who ranked 10 for all presentations. The ranking are separated by the self-evaluation (team members ranking themselves), customer evaluation, peer evaluation (other students ranking the teams), and overall ranking. The average of all ranking for each team based on the project is as follows:

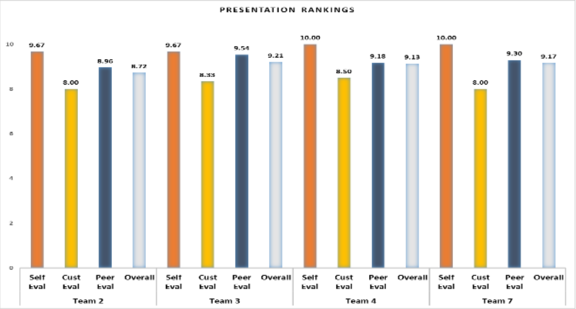


Figure 1 Presentation Ranking for project 1

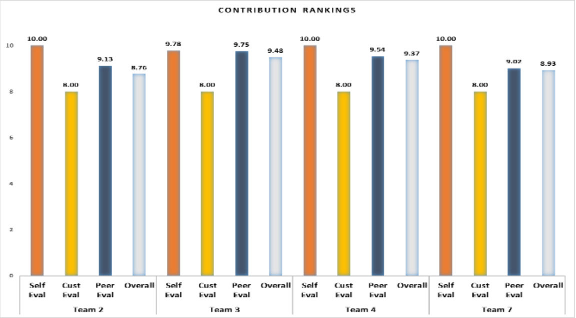


Figure 2 Contribution ranking for project 1

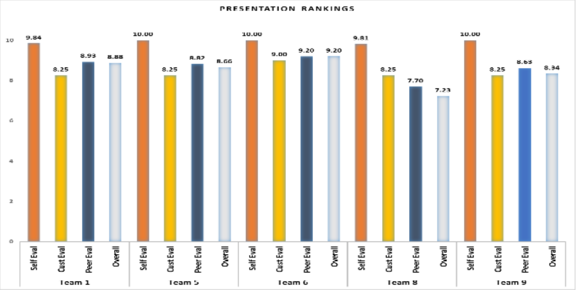


Figure 3 Presentation ranking for project 2

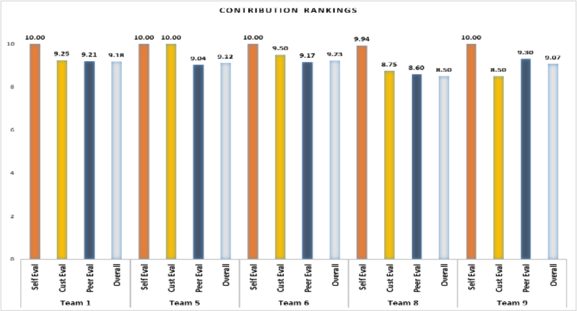


Figure 4 Contribution ranking for project 2

Conclusion

There are several other ELOs that were verified and validated using the project in this course but only a few are shown in this paper. The data collected demonstrates the assessment mostly related to the 3 specified ELOs.

For the first project, the data indicates that all the teams had a higher self-evaluations for their contribution and their presentation than their peers. The customer had the lowest evaluations for both presentation and contribution on all teams.

For the second project, self-evaluation was the highest ranking in both presentation and contribution. However, the customer had the lowest presentation ranking for all teams except one. The contribution is mixed among customer and peer evaluations.

The data supports the fact that the teams worked relatively well together but the expectation of the customer was not fully met.

References:

[1] Global Harmonization Task Force - Quality Management Systems - Process Validation Guidance (GHTF/SG3/N99-10:2004 (Edition 2) page 3

[2] IEEE. "IEEE Guide--Adoption of the Project Management Institute (PMI®) Standard A Guide to the Project Management Body of Knowledge (PMBOK® Guide)--Fourth Edition". p. 452.

[3] R. Sargent, “Verification and Validation of simulation models”, journal of Simulation (2013) 7, 12–24. doi:10.1057/jos.2012.20; published online 7 Dec 2012

[4] ISO website (http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm)

[5] US Department of Education (http://www2.ed.gov/about/what-we-do.html)

[6] Association of American Colleges and Universities – Learning Outcomes Assessment (http://leap.aacu.org/toolkit/learning-outcomes-assessment)

Analysis of the relationship between previous ideas about the physics concepts and a poor student's performance in classroom (FECS'14)

Guzmán R. Miguel A. [1], Rosete F. Juan C. [2], Estrada R. Felipe [3]

1 Computer Engineering Department, Technological Institute of Queretaro,

2 Mechanical Engineerig Department, Technological Institute of Queretaro
Av. Tecnológico S/N Esq. M. Escobedo, PC 76000, Queretaro, Qro, México
jc_rosete@yahoo.com.mx

3 Computer Engineering Department, Technological Institute of Queretaro

Abstract. - *This paper exposes the relationship between common misconceptions of the physics of movement in the classroom and the serial of previous ideas developed by engineering students along their life with the forced consequence of failed notes and further problems with lectures related with those topics.*

This paper shows the results of a research developed into the Technological Institute of Queretaro in order to identify typical misconceptions about the bodies' behavior in static and dynamic conditions which could produce problems to develop models of our environment. With the obtained information a software system is developed in order to conduct a serial of computer experiments that confronts the student's ideas with physics laws in order to force cognitive conflict within the student, leading it to the building of new concepts and a better understanding of the lecture.

Keywords: previous ideas, physics, cognitive models

1 Introduction

Desertion and low notes are typical problems that several countries around the world experiment continuously in their day by day educative process. These conditions are normally evaluated as a way to identify the economical amount of resources that every year the country loses for this concept and are the basis to plan strategies oriented to improve the activity into and out of the classroom.

A lot of students experiments certain level of afraid when a lecture of mathematics must be coursed, and in a low number of cases of study, the student considers self-sufficient or with the competence to solve properly a math exercise.

A different condition is identified when the topic of study is the physics, since a large number of students in their own opinion consider this basic science easier than math and with a higher dominion of the topics related to the discipline. Nonetheless this point of view from the students, the rate of failing is close to the 50% of students in all engineering programs [1].

This condition is the result several factors acting simultaneously including large number of lectures per day, the cultural influence of mass communication media, a lack of criteria to discern the quality of information that the individual can find in the internet lots of misconceptions learned along the student´s life. All these can be conditions that could lead a student toward low performance in classroom or to fail a course, because they are not able to establish logical connections between the learned theory and their real life experiences.

In the particular situation of physics lectures three elements are common: a low level in math, a low development of abilities to establish connections between theoretical concepts and real life situations [2] and previous ideas about a topic that act as an inhibitor of the learning process [3].

In order to identify what kind of wrong previous ideas our students of engineering have, it was applied the “Force Concept Inventory” (FCI), it is a test with multiple options developed in 1992 whose main objective is to evaluate if the student is capable to do a right answer in physics problems nonetheless the great quantity of failed concepts learnt for the individuals along their entire lives.

2 Misconceptions detection

2.1 Structure of the test

The first step was to apply a serial of tests to students of computer engineering with 10 questions selected from the “Force Concept Inventory” [4], about the bodies´ movement understanding.

The test is designed explaining a possible situation of a body under movement and the student must think about the logical answer that could join the situation with the information and experiences that previously the student has obtained.

Every one of the questions contain the description of one of such situations, most of the times a picture that shows several options to describe the behavior of the body and a total of 5 options to choose one of them as the right answer.

The next example can explain the nature of the questions and the level of misunderstanding about the topic in physics.

“A canon ball is launched at 0 degrees. What option can describe the movement of the ball mass after it is out of the canon?”

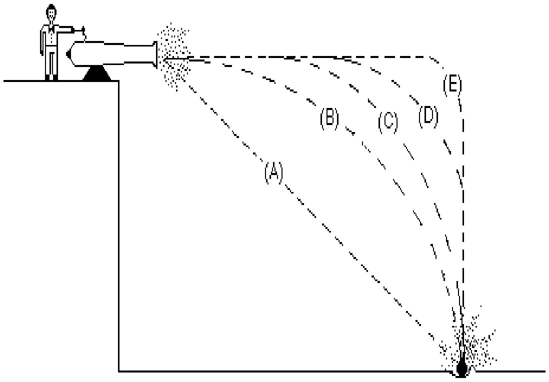


Figure 1 Question example

With the following results

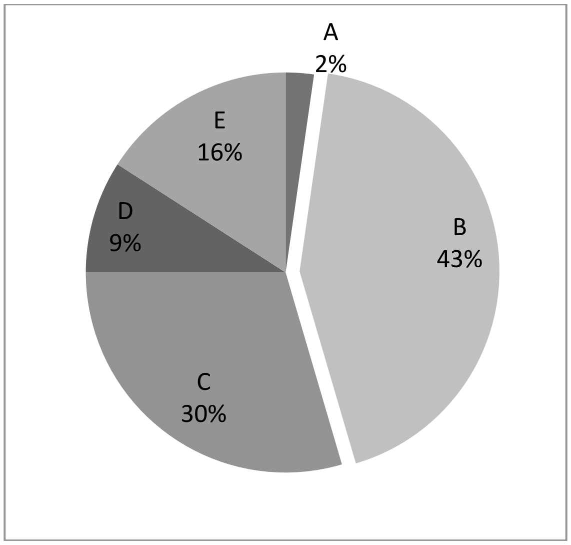


Figure 2 Question number 4 selections

The graphic shows the number of students (43 %) that chose the correct option (B), it corresponds to one parabolic movement given that the combination of movement vectors along x axis with constant velocity and the vector along Y axis with accelerated conditions because the action of the gravity.

It is important to check the high number of students that chose option E nonetheless the impossibility of such movement condition. This number of responses can suggest that there are several involved factors that lead the student to consider as valid possibilities those that normally in the real life are not considered as a typical response of a system.

2.2 Validating proposal

At this point it was necessary to validate if the previous ideas can influence the mental procedures of one individual to consider possible movement conditions that the serial of experiences in our life have shown that are not valid.

Another question from the FCI requests the movement behavior of a body that is dropped from an airplane in flight. The 5 options show different combination of movement but only option D is correct.

The next 2 figures define the quest conditions and the percentage of students that selected it. The result show several difficulties for the individuals to understand the behavior of bodies under movement.

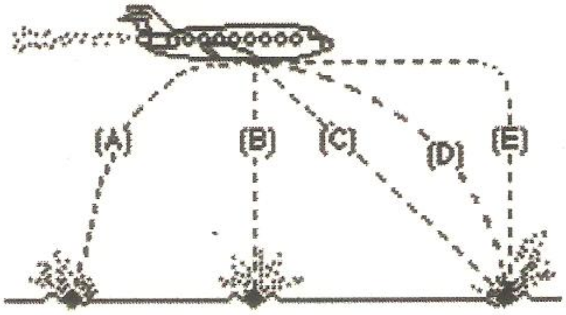


Figure 3. Dropped body movement

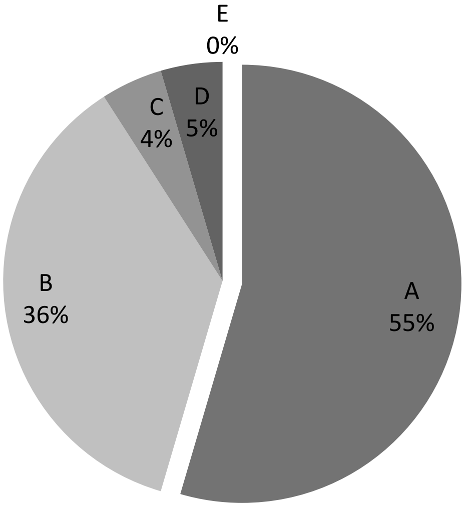


Figure 4 Plane question results. This test was made to students that have not studied the formal physics course

With this information it is obvious that a large number of students have troubles to establish proper connections with a proposed condition and the physics laws.

2.2.1 Knowledge in physics

Another hypothesis formulated to define the potential reason of this kind of mistakes is that most of the students in second semester have not matriculated the course of physics and it can be one reason of the obtained results.

Again, the same test was applied to different groups of students of computing engineering but now with the difference that they had completed and covered the conditions to validate the lecture. This condition should improve the number of right answers in the same test, because of the serial of experiences and academic resources applied into and out of the classroom.

The next graph shows the result of the plane question with students that had approved the physics course:

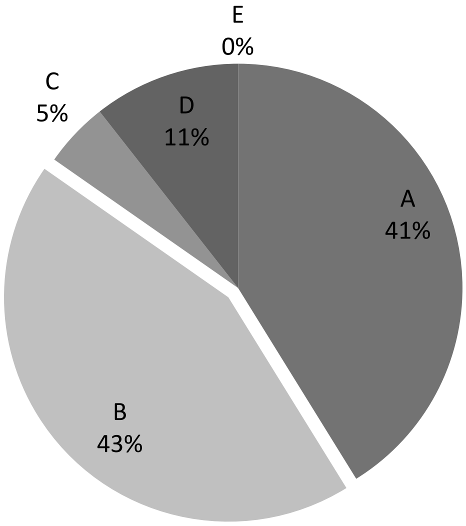


Figure 5 Plane question and results with students of higher levels after the physics course

It means that the impact of the lectures in the understanding of physics laws is poor, because only 11% of students concluded that D was the right option in comparison with the original 5%, and the number of students that selected option B increased in a7%.

The total test let us to conclude that, the information provided in a typical program of physics is not enough to change the understanding level of the topic and it is sure that this condition will affect the performance of the student in consecutive courses given that serial of mistakes that the individual is carrying on.

In order to identify the more important reasons of this condition a secondary research was done. It was found that most of the students that were wrong in the first study have a great influence of the concepts and situations that the mass media such as TV, video games and movies show.

Figure 6 The roadrunner® illustrates the impossible cartoon physics

® WB Looney tunes, all rights reserved

2.2.2 Applying software and hardware to develop experiments

Most misconceptions about physics are about science in general, but these are most obvious in the case of physics, because of all the sciences it is the most foreign to most people, because it seems to relate the least to everyday events. In fact, the opposite is true, since physics applies to everything, while the other sciences are more specialized. Physics school experiments play a crucial role in motivation of students in physics education; these experiments form a crucial element in effective cognitive motivational teaching techniques.

One of the most cheap and powerful tools available today for measuring results in an experiment is the Arduino® board, a cheap readily available microprocessor board designed right from the start to let non-engineers make use of computer-control for projects. According to the official web site (www.arduino.cc), "Arduino® is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software."

The advantages to guide a test and experiments under Arduino® platform is the possibility to validate the behavior of a body under movement applying sensors that can measure the variables that are being studied, and at same time the student interacts with electronic systems increasing by this way the point of view and synergetic behavior of the future engineers.

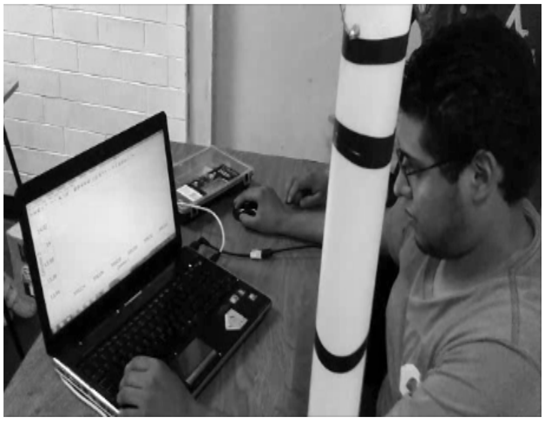


Figure 7 Programming the Arduino board

Arduino® can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. This can be used to make a powerful recipe for engaged learning: Show students how to command actions in the physical world – make lights blink, sounds sound, motors move, robots roam, sensors sense. Combine this concrete act of control over physical objects activity with the abstract power of programming – show students how to make those lights blink in response to the sounds, make the sensors guide the motors.

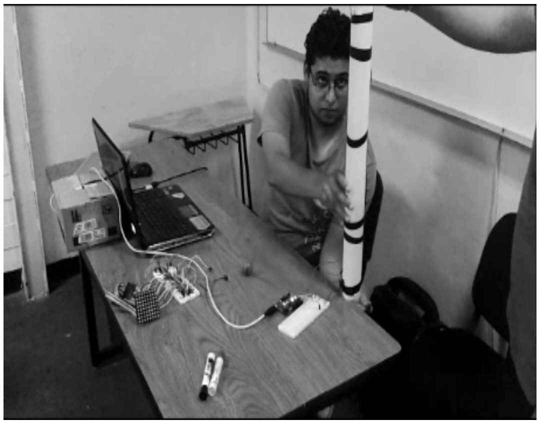
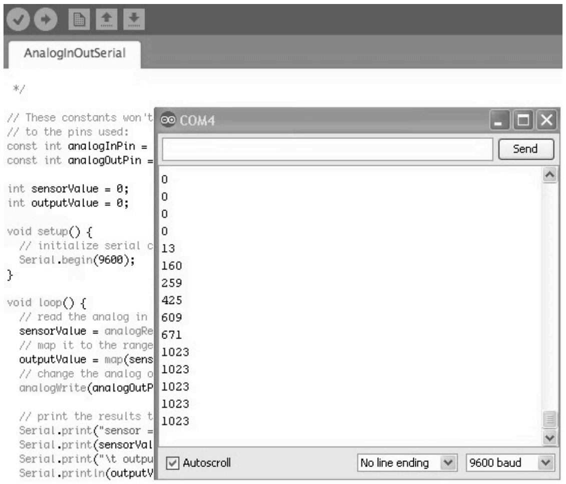


Figure 8 Free fall systems

The students at the Technological Institute of Querétaro designed and implemented several experiments to show the behavior of a free falling object, using basic sensors to send the signal to an Arduino board in order to measure the speed of a body under the acceleration of gravity. They programmed the required software to make sense of these data, and proceeded to draw conclusions after comparing these results with their previous ideas about this phenomenon, which often led them to a cognitive conflict.



```
 */

// These constants won't
// to the pins used:
const int analogInPin =
const int analogOutPin =

int sensorValue = 0;
int outputValue = 0;

void setup() {
  // initialize serial c
  Serial.begin(9600);
}

void loop() {
  // read the analog in
  sensorValue = analogRe
  // map it to the range
  outputValue = map(sens
  // change the analog o
  analogWrite(analogOutP

  // print the results t
  Serial.print("sensor =
  Serial.print(sensorVal
  Serial.print("\t outpu
  Serial.println(outputV
```

```
0
0
0
0
13
160
259
425
609
671
1023
1023
1023
1023
1023
```

No line ending
9600 baud

Figure 9 Source code based upon the Wiring programming framework for Arduino, and obtained data from the free fall experiment

2.2.3 Obtained results

The obtained results with the tool were totally satisfactory. All the groups of students were involved in the solution of an engineering problem that required the development of a serial of abilities and competences, not only in physics, because it was required to practice with sensors, programing and to understand the ways to stablish the interface with a computer to get the results of the experiment.
After this experience the group (not the same than the original data) was requested to answer the test and the graph can show us the difference when a tool able to validate or refuse our ideas and concepts is applied.

This result tell us than it is possible to improve the level of understanding in a topic if the proper didactictools are applied.

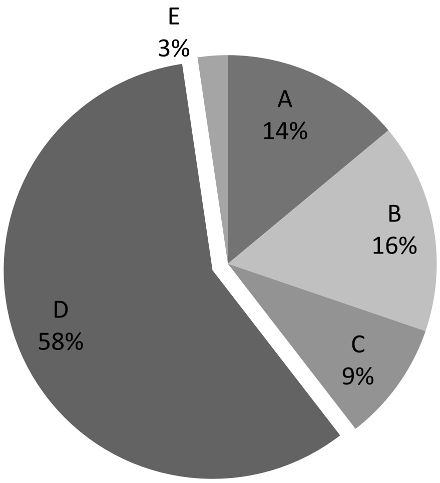


Figure 10 Number of right answer D, after applying the Arduino board.

3 Conclusions

This time is plenty of electronic resources that allow the students to get information about all kind of topics; unfortunately, more information does not mean more understanding of the topics to the students.
It is important to the teacher into the classroom to adapt the new technologies with the learning process looking for a way to improve the performance of the students

4 References

1. Jimenez G. M. A. (2011) Rate of non satisfactory results into a physics course
2. Stahl, G., Koschmann, T., Suthers D, (2006) Computer – supported collaborative learning: An historical perspective. Cambridge handbook of the learning sciences, pp 409 – 426
3. Kind, V. (2004) Beyond appareances: Students´ misconceptions about basic ideas, 2nd edition.
4. Hestens, D., Wells., Swackhammer, G. (1992), Force Concept Inventory. The physics teacher, 30, pp. 141 - 158
5. Hay D., Kinchin, (2008) Making learning visible: the role of concept mapping in higher education, Studies in higher education, 33 (3), pp. 295 - 311

Integrating Culture Intelligence into Computer Science

Zhao Xin Wu[1], **Li ZHOU**[2]

[1]Computer Science Department, University of Quebec in Montreal, Montreal, Canada
zhao_xin_wu@hotmail.com
[2]School of Electronic and Control Engineering, Chang'an University, Xi'an, P.R. China
47599053@qq.com

Abstract - *Research on Cultural Intelligence provides a new perspective and promising means of reducing intercultural conflicts and obstacles. Up until now, research in this domain has used traditional methods and has relied mainly on questionnaires to manually evaluate the Cultural Intelligence of individuals. This research aims to invent a Cultural Intelligence evaluation computational model which implements the model in an expert system, in order to evaluate precisely the Cultural Intelligence of individuals and organizations through the use of Artificial Intelligence technology. The purpose of this system is to support users in solving the intercultural adaptation problems that they face in various authentic situations. As a 'culturally aware' intelligent evaluation system, this system can be used to evaluate individuals and organizations in Cultural Intelligence training, and provides specific suggestions to improve their weaknesses in the corresponding area. This is a particular importance in modern learning theories.*

Keywords: Cultural Intelligence, Evaluation System, Fuzzy logic, Artificial Neural Networks, Hybrid Neuro-Fuzzy Intelligent System

1 Introduction

In the context of globalization, individuals are required to make culturally relevant decisions and to show their competence in culturally diverse workplaces. Once in a foreign country, some individuals can easily and successfully integrate into the local culture. In contrast, others find it difficult or impossible to adapt to a new context. Why does this difficulty exist? What do people need in order to rapidly and effectively adapt to a new culture? What can be done to improve people`s capacity to adapt to a different culture?

In recent years, more and more cultural experts and researchers have shown great interest in intercultural communications. They believe that a person`s extent and development in their capacity of cultural adaptation are more important, more accurate and more effective than simply transmitting different cultural knowledge to a person. This adaptation ability in different cultures to a person is called Cultural Intelligence (CQ) [1]. Thus, the CQ is presented as a new phenomenon of effective solving problems of cultural conflicts. Since it was proposed, organizational psychology and human resource management have given it much attention.

However, studies on the CQ are currently performed with manual processing. In this study, we propose at first a CQ computational model, and then we implemented the model in an expert system based on an innovative breed of AI technologies, to evaluate individuals and to give them recommendations to improve their CQ, so that they can appropriately interact with different cultures. This system has three main functions: one function is to precisely assess an individual`s CQ skill level; the other is to provide concrete recommendations; while the third is to facilitate researchers` work and to equip them in their research concerning CQ. The system is considered highly intelligent due to its wealth of knowledge, openness, scalability, flexibility, adaptability, and capability to self-learn. As a result of these qualities, the system allows better interaction and more effective aid in the evaluation process so as to improve users' cultural skills in different cultural settings

2 Cultural Intelligence

In the research literature, Earley and Ang presented CQ as a reflection of people's ability in a new cultural context to collect and process information, to make judgments, and to take effective measures to adapt [2]. Earley and Mosakowski later redefined CQ as what a manager refers to as his ability to deal calmly with different cultures [3]. Early and Mosakowsi stressed that people with a relatively high CQ level often appear at ease in new situations. They understand the subtleties of different cultures, so they can avoid or resolve conflicts early.

Peterson interprets CQ in terms of its operation [4]. He believes that, for the concept of CQ, the definition of 'culture' is compatible with the cultural values of Hofstede and their five main dimensions [5]. Peterson also describes CQ as the communicative capabilities which improve working environments. In other words, all the workers have the ability to communicate efficiently with customers, partners and colleagues from different countries in order to maintain harmonious relationships.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Ang and al. define CQ as the conceptualization of a particular form of intelligence based on the ability of an individual to reason correctly in situations characterized by cultural diversity [6]. They suggest that CQ is a complementary form of intelligence which may explain the ability to deal with diversity and function in a new cultural setting.

Earley and Ang are pioneers in the development of CQ concepts by having created a three-dimensional model in 2003. Ang and Van Dyne later refined this concept to consist of four dimensions structure rather than three. This structure has been widely used in the following cultural research and studies. They paid special attention to how a culturally diverse environment works [6].

1) Metacognition refers to the cognitive ability of an individual to recognize and understand appropriate expectations in different cultural situations. It reflects the mental processes that an individual uses to acquire and understand cultural knowledge.

2) Cognition is a person's knowledge of the standards, practices and conventions in different cultures which he/she acquired from education and personal experiences.

3) Motivation refers to the motivation of an individual to adapt to different cultural situations. It demonstrates the individual’s ability to focus his/her attention and energy on learning and practicing in culturally diverse situations.

4) Behavior is defined as an individual’s ability to communicate and behave with cultural sensitivity when interacting with people of different cultures. It represents a person’s ability to act and speak appropriately (i.e., use suitable language, tones, gestures and facial expressions) in a given culture [7].

3 Cultural Intelligence Computational Model

3.1 Data Acquisition

Kon et al. [7], Ang et al. [8], and Ang et al. [6] developed a self-assessment questionnaire which has 20 items that measure CQ. This questionnaire was used to collect data for studies on the capabilities of the test subjects regarding their cultural adaptation capacity. This questionnaire is generally divided into four sections: metacognitive, cognitive, motivational and behavioral. For example, one of the items is "*I am conscious of the cultural knowledge I use when I interacting with people with different cultural backgrounds*." Van Dyne et al. [9] developed a version of the questionnaire from the point of view of an observer. It is also based on the 20 items of Ang et al. [6] in order to measure the CQ of individuals. The questionnaire was adapted from each item of the self-assessment questionnaire to reflect the assessment made by an observer rather than the user himself. For example, the item of the questionnaire shown above changes from: "*I am conscious of the cultural knowledge I use when…*" to "*This person is conscious of cultural knowledge he / she uses when* ...." As explained by Van Dyne et al. [9], these questionnaires allow for the effective assessment of CQ in practical applications. Thus, we adapted the self-assessment questionnaire of Ang et al. [6], along with the observer questionnaire by Van Dyne et al. [9] to measure CQ in order to integrate the evaluation functions offered by our model.

3.2 AI Technologies

The model is made up of a combination of two AI technologies, which makes it possible to achieve a CQ system capable of providing better evaluation and improving the adaptability of individuals in intercultural settings. The two main technologies employed are: 1*) Hybrid Neuron-Fuzzy Technology*: fuzzy logic is used when the natural language describing the concepts of CQ is ambiguous and imprecise; Although the fuzzy logic technology has the ability and the means of understanding natural language, it offers no mechanism for automatic rule acquisition and adjustment. The Artificial Neural Network (ANN) technology is an adaptive system that can be formed and adjusted from a set of samples. The ANN can handle new data input with the generalization of acquired knowledge. Thus, hybrid neuro-fuzzy is used as it combines and makes use of the advantages and power of fuzzy logic and ANN, which are complementary paradigms. *2) Expert System (ES)*: A knowledge base which allows knowledge engineering to transform expert cultural knowledge using fuzzy logic, such as the basic knowledge of law, economics, is also required. The ES was, therefore, chosen as the second technology. An ES cannot learn, but it can explain its reasoning. The combination of these technologies allows us to attain the technical aspects, as per our system development requirements.

3.3 Creating Fuzzy Sets

All the fuzzy sets come from the CQ domain. A practical approach to form fuzzy sets is used in our model. We take the first dimension of CQ *Metacognition* as an example (See section 2), when *X* is the universe of discourse of metacognition, and its elements are denoted as *x*, the fuzzy set *Metacognition (MC)*, is part of the universe *X* and is defined by the function $\mu MC(x)$ as a function of membership in the set *Metacognition*. This equation is expressed as: $\mu MC(x): X \rightarrow [0,1]$. For every element *x* of universe *X*, the membership function $\mu MC(x)$ equals the degree to which *x* is an element of set *Metacognition*. This degree, with a value between 0 and 1, represents the level of membership of element *x* to set *Metacognition*.

As our model is also based on ANN technology, our second approach to form fuzzy sets is to learn from the data available to the CQEES, and then to derive the fuzzy sets automatically.

3.4 Fuzzy Rules and Operations of Fuzzy Sets

The idea of linguistic variables is one basis of the fuzzy set theory. A linguistic variable is a fuzzy variable. For example, when we say "*the Cultural Intelligence is high*," it means that the linguistic variable of CQ takes the linguistic value *high*. Thus, our linguistic variables are used in fuzzy rules in the model. For example:

Rule x:

IF	Metacognition is high AND cognition is high AND motivation is high AND behavior is high
THEN	CQ is high

Rule y:

IF	Enjoys interacting OR confident to socialize with people from other cultures OR likes living in different cultural settings OR confident in himself OR handles stress well in cultural adaptation
THEN	Motivation is high

The operations of fuzzy sets used in our model are the *Intersection* (AND) and *Union (*OR*)*. For example, the fuzzy operation used to create the *Intersection* of two fuzzy sets A and B is as follows:

$$\mu A \cap B(x) = min[\mu A(x), \mu B(x)] = \mu A(x) \cap \mu B(x), where\ x\ \in \mathrm{X} \quad (1)$$

The operation to form the fuzzy *Union* of two fuzzy sets A and B is as follows:

$$\mu A \cup B(x) = max[\mu A(x), \mu B(x)] = \mu A(x) \cup \mu B(x), where\ x \in X \quad (2)$$

Considering two fuzzy sets of *High Metacognition* and *Medium Metacognition* of a person (between 0 and 10):
High metacognition = (0.1 / 1, 0.7 / 7, 0.9 / 9);
Medium metacognition = (0.5 / 1, 0.3 / 7, 0.1 / 9)

According to equation (1), the *Intersection* of these two sets is:

High metacognition ∩ *Medium metacognition* = (0.1/1, 0.3/7, 0.1/9)

According to equation (2), the *Union* of these two sets is:

High metacognition U *Medium metacognition* = (0.5/1, 0.7/7, 0.9/9)

3.5 Applying AI Technology to Model

This model is based on the theory of Ang et al. [6] and its four dimensions of CQ: metacognition, cognition, motivation and behavior, as well as the other most important aspects. We agree that the four dimensions are critical factors that can help individuals and organizations to overcome cross-cultural challenges. Thus, we applied the four dimensions in order to integrate the elements required to respond to the CQ evaluation cultural knowledge acquired in cross-cultural activities.

The neuro-fuzzy model is similar to a multilayer neural network. The model uses a technique called the fuzzy inference method by Mamdani [10]. Fig.1 illustrates in the model an example of the application of the technique by using triangular sets. We define that the fuzzy system as having four inputs: *metacognition*, *cognition*, *motivation*, and *behavior*, and as having one output: CQ. For example, input *metacognition* is represented by metacognitive fuzzy sets 1,2,3,4,5,6; output CQ is represented by fuzzy sets CQ 1, 2,3,4,5 and 6.

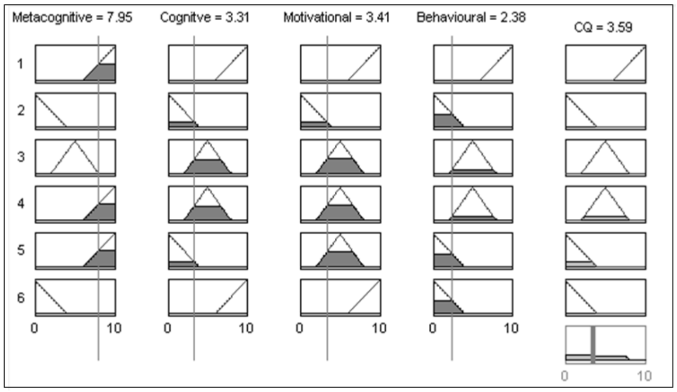


Figure 1. Example of the fuzzy inference system of Mamdani using triangular setsCQEES

Fig. 2 shows the neuro-fuzzy model that corresponds to this fuzzy inference system. It is represented with a neural network composed of five layers in the model. Each layer of the neuro-fuzzy inference model is associated with a particular step in the fuzzy inference process (See Fig.1). It also has four inputs: *metacognition* (MC), *cognition* (C), *motivation* (M) and *behavior* (BEH), and has one output: CQ

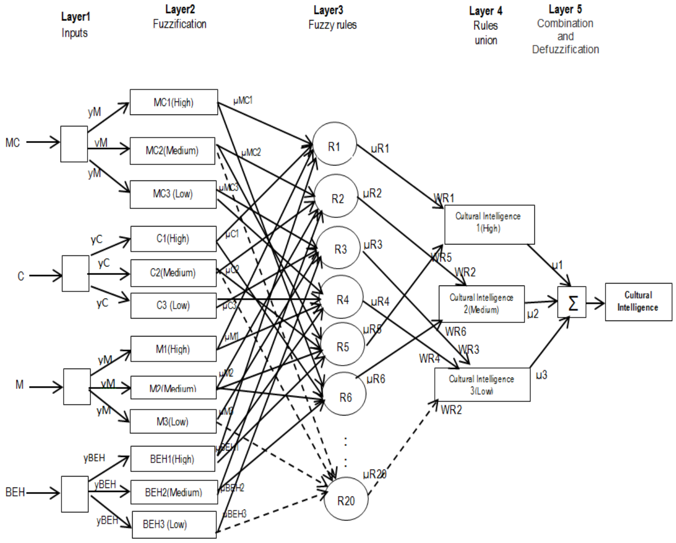


Figure 2. Five layers CQ computational model

Layer 1 - Input: No calculation is made in this layer. Each neuron corresponds to an input variable. These input values are transmitted directly to the next layer.

Layer 2 - Fuzzification: Each neuron corresponds to a linguistic label (e.g., high, medium and low) associated with one of the input variables in layer 1. In other words, the connection of the output, representing the inclusion value which specifies the degree to which the four input values belong to the neuron's fuzzy set, is calculated in this layer.

Layer 3 - Fuzzy Rule: The output of a neuron at level 3 is the fuzzy rules of CQ. Each neuron corresponds to one fuzzy rule. The neuron receives as input from the Fuzzification neurons. Neuron R1 represents Rule 1 and receives input from the neurons MC1 (High) and C1 (High). The weights (WR1 to WRn) between layers 3 and 4 are the normalized degree of confidence of the corresponding fuzzy rules. These weights are adjusted when the model is trained.

Layer 4 - Rule Unions (or consequence): This neuron has two main tasks: 1) to combine the new precedent of rules, and 2) to determine the output level (High, Medium and Low), which belongs to the CQ linguistic variables. For example, $\mu R1$, $\mu R5$ are the inputs of CQ1 (High), and $\mu 1^{(4)}$ is the output of neuron CQ1 (High).

Layer 5 - Combination and Defuzzification: This neuron combines all the consequence rules and, lastly, computes the crisp output after Defuzzification. The composition method "sum-product" [11] is used. This method represents a shortcut of the Mamdani-style inference calculation. It computes the outputs of the membership functions defined by the weighted average of their centroids. The calculation formula of weighted average of the centroids of the clipped fuzzy sets CQ 3(Low), 2 (Medium) and 1(High) are calculated as shown in Fig. 3:

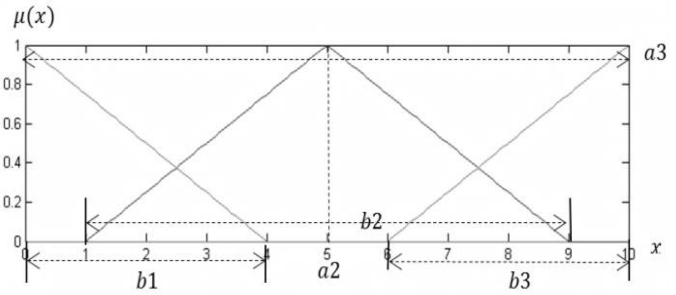


Figure 3. General CQ Fuzzy Sets

$$y\ (Cultural\ Intelligence) = \frac{\frac{1}{3}b_1^2\mu 1 + a_2 b_2 \mu 2 + \left(a_3 - \frac{1}{3}b_3\right) b_3 \mu 3}{b_1\mu 1 + b_2\mu 2 + b_3\mu 3} \qquad (3)$$

Where *a2* is the center and *a3* is the end of the triangle. *b1*, *b2* and *b3* are the widths of fuzzy sets which correspond with CQ 3, 2 and 1.

3.6 The Implementation of the Model into an Expert System

This section presents the conceptual structure of our system. We combined two intelligent technologies: hybrid neural-fuzzy and expert system. As a hybrid intelligent system, it provides comprehensive and global solutions and forms a system of rules capable of adapting to a multicultural environment. This is the context in which the system was born. The hybrid neural-fuzzy technology ensures that the system is capable of reasoning and learning in an uncertain and imprecise cultural environment. The expert system, meanwhile, uses the knowledge of cultural experts and inference procedures in order to solve difficult problems normally requiring human expertise in the CQ domain. This synergy improves adaptability, fault-tolerance robustness and speed of system. In fact, the name of our system is: Cultural Intelligence Evaluation Expert System (CQEES). Fig. 4 illustrates the general structure of the CQEES. The structure includes four main modules:

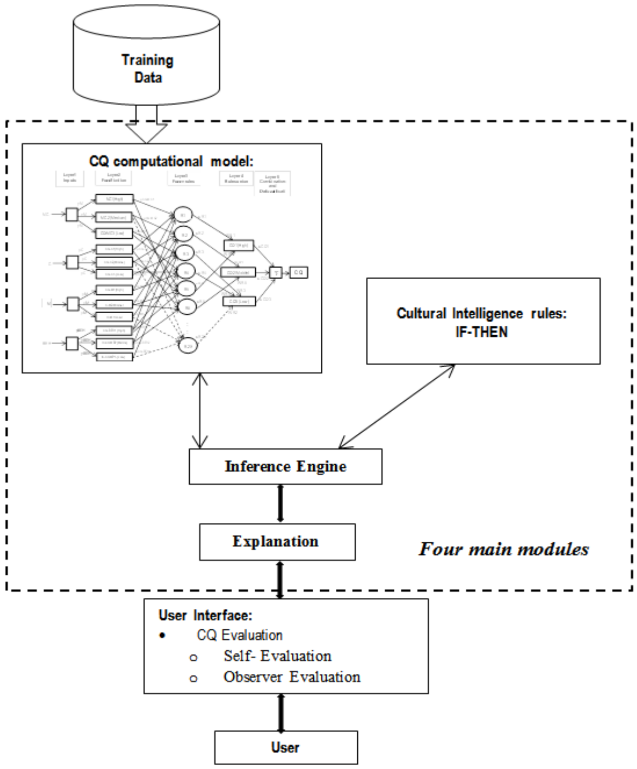


Figure 4. Complete Structure of CQEES

1) *The CQ Computional Model* contains CQ knowledge that is useful for solving cultural problems. The model in this structure is represented by the trained neural-fuzzy network. This module supports all the cultural evaluation steps in the system. It connects with *Training Data* unit. *Training Data* are a set of training examples that are used for training the network during the learning phase.

2) *The Cultural Intelligence IF-THEN Rules* examine the CQ knowledge base, which is represented by the computational model, and produce rules which are implicitly "buried" in the neuro-fuzzy network.

3) *The Inference Engine* is the core of the CQEES. It controls the flow of cultural information in the system and initiates inference reasoning from the computational model. It also concludes when the system has reached a solution.

4) *The Explanation* clarifies to the user why and how the CQEES has achieved the specific CQ evaluation results. These explanations include the conclusion, advice and other facts required for deep reasoning.

The CQEES possesses generic CQ and is not specific to a particular culture, such as that of the United States or China. The system shows big capabilities of cultural adaptation by modeling the human decision-making process in situations characterized by cultural diversity. Furthermore, due to its intricate cultural schemas and analytical abilities, the system can help users identify and understand key issues in CQ judgment. It also gives them the corresponding explanations. In this research, C/C++ is chosen as the programming language.

4 System Architecture

This architecture normally allows for an approximate and dynamic representation of the CQEES in terms of the storage and flow of information in the system. Fig. 5 shows the architecture. The system is generally regarded as beginning with an information input number and ending with a response shown to the user.

The three main parts of the architecture are: (1) *The Input Process* presents information (a number between 0 to 10) which expresses a user's answer via the input of the user interface. Through the Identify Unit to distinguish which CQ dimension the user wants to answer the questions; the Filter and Classifier module takes the inputted information, classifies it, and filters what is not useful for analysis in the next steps; (2) *The Analyse Process* is a neural network with fuzzy inference model capabilities. This module has four sub-modules: Metacognition (MC), Cognition (C), Motivation (M) and Behavior (BEH). Each module in this section must interact with each other so that each sub-module can adjust its result. This interaction gives a more complete and effective result before continuing to the *Recommendations* section. In this section, the system can be trained to develop IF-THEN CQ fuzzy rules and can determine membership functions for input and output variables.

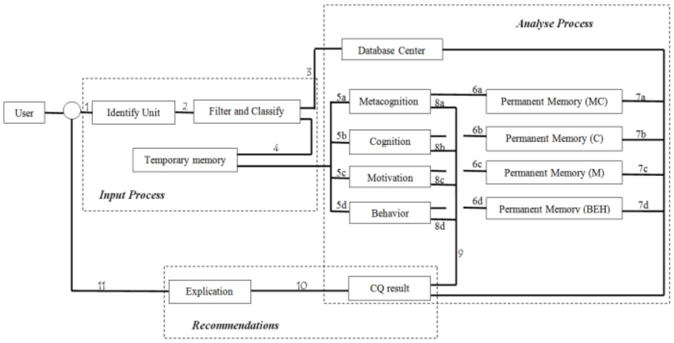
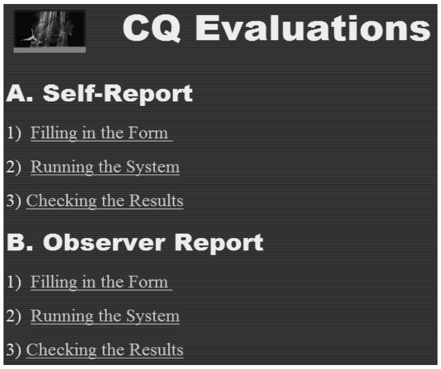


Figure 5. Architecture of CQEES

(3) *The Recommendation* section has the generalized results from the *Analyse Process* section, and explains the results of evaluation to users in natural language and provides suggestions. The explanations are sent to the User interface.

5 System Results

The system applies measurement strategies which are represented by 20 questions that human cultural experts used during their CQ evaluations. The system provides important insights about personal capabilities and information on the user's own CQ in cultural diversity situations. Users could get two evaluations (self or observer evaluation) on the 20-itemed questionnaires so as to compare their results. Organizations could also use CQEES (both self- and observer -evaluations) to evaluate for expatriate purposes employees who may be well-adapted. Therefore, the following details explain how users can get two evaluations (self and observer evaluations) using the 20-item questionnaires (see the interface of the CQEES in Fig. 6).



CQ Evaluations

A. Self-Report

1) Filling in the Form

2) Running the System

3) Checking the Results

B. Observer Report

1) Filling in the Form

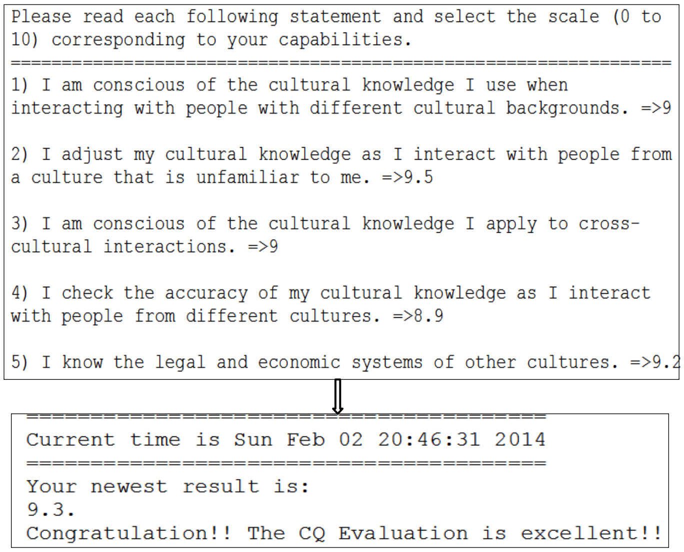
2) Running the System

3) Checking the Results

Figure 6. Interface of CQEES

For example, two different results of the self-evaluation questionnaire that evaluate the user's CQ are presented in the CQEES as follows:

Result 1: After inputting the answers to the 20 items in the CQEES, the system provides the feedback. If a user's evaluation achieves a high score (e.g.: more than 8), the system shows the following message in Fig. 7:

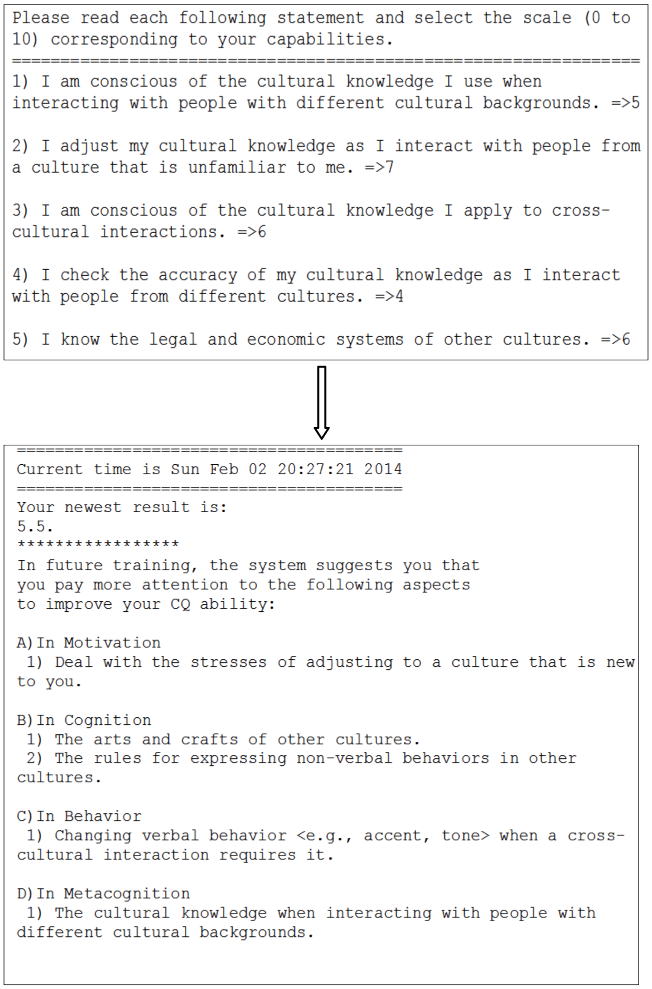


```
Please read each following statement and select the scale (0 to
10) corresponding to your capabilities.
================================================================
1) I am conscious of the cultural knowledge I use when
interacting with people with different cultural backgrounds. =>9

2) I adjust my cultural knowledge as I interact with people from
a culture that is unfamiliar to me. =>9.5

3) I am conscious of the cultural knowledge I apply to cross-
cultural interactions. =>9

4) I check the accuracy of my cultural knowledge as I interact
with people from different cultures. =>8.9

5) I know the legal and economic systems of other cultures. =>9.2
```

```
========================================
Current time is Sun Feb 02 20:46:31 2014
========================================
Your newest result is:
9.3.
Congratulation!! The CQ Evaluation is excellent!!
```

Figure 7. One part of the self-evaluation result in CQEES (high score more than 8)

Result 2: The system presents feedbacks when a user gets a score lower than 6. In addition, it accordingly gives useful suggestions for personal self-development as required. This process permits the system first to evaluate the user so as to identify their problems in the CQ domain. The system next offers a recommendation to user based on the results of the evaluation. The system presents some recommendations in Fig. 8:



```
Please read each following statement and select the scale (0 to
10) corresponding to your capabilities.
=============================================================
1) I am conscious of the cultural knowledge I use when
interacting with people with different cultural backgrounds. =>5

2) I adjust my cultural knowledge as I interact with people from
a culture that is unfamiliar to me. =>7

3) I am conscious of the cultural knowledge I apply to cross-
cultural interactions. =>6

4) I check the accuracy of my cultural knowledge as I interact
with people from different cultures. =>4

5) I know the legal and economic systems of other cultures. =>6
```

```
========================================
Current time is Sun Feb 02 20:27:21 2014
========================================
Your newest result is:
5.5.
*****************
In future training, the system suggests you that
you pay more attention to the following aspects
to improve your CQ ability:

A)In Motivation
 1) Deal with the stresses of adjusting to a culture that is new
to you.

B)In Cognition
 1) The arts and crafts of other cultures.
 2) The rules for expressing non-verbal behaviors in other
cultures.

C)In Behavior
 1) Changing verbal behavior <e.g., accent, tone> when a cross-
cultural interaction requires it.

D)In Metacognition
 1) The cultural knowledge when interacting with people with
different cultural backgrounds.
```

Figure 8. One part of the self-evaluation result in CQEES (lower than 6)

6 Validation of CQEES

Three cultural experts have validated our computational CQ model and the CQEES system. This validation ultimately reflects the consistency between the real world and the artificial CQEES system. The CQEES system was also tested with two hundred people by measuring their CQ value. The effectiveness and robustness of the system is evaluated by carrying out a regression analysis on these data. The correlation coefficient *R* between the system outputs and the corresponding experts' desired values is calculated, *R=0.9001*. The Fig. 9 shows the results of the analysis. After training the system with desired data, the final *R* is close to 1.

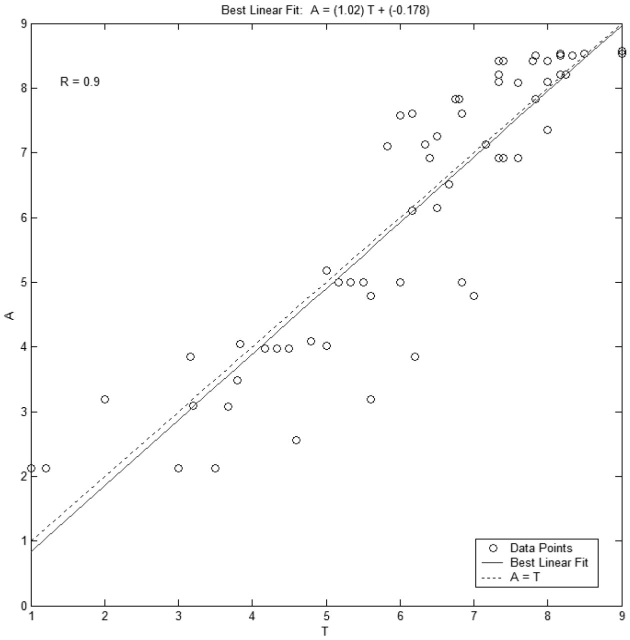


Figure 9. The regression analysis of two hundred people CQ values

Table 1 shows the results between manual evaluation and CQEES evaluation. We observed that the CQEES produces different results than manual evaluation for each respondent.

Table 1: Comparing the differences between manual evaluation and CQEES

Users	Questions 1-20																			
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q 10	Q 11	Q 12	Q 13	Q 14	Q 15	Q 16	Q 17	Q 18	Q 19	Q 20
User1	0	0	2	2	4	5	6	6	6	7	7	7	7	8	8	8	8	9	10	10
User2	9	9	8	5	4	4	6	4	2	2	10	9	10	10	9	9	8	9	5	6
User3	8	7	7	9	4	6	3	3	5	5	5	4	3	5	6	7	7	8	8	10
User4	10	8	9	6	7	6	9	5	5	7	10	10	10	10	10	6	0	0	0	0

Evaluations	Average	CQEES
User1	6.0	1.0
User2	6.9	3.7
User3	6.0	4.3
User4	6.4	1.2

The evaluation results made by CQEES are similar to a cultural expert, and these results are confirmed by three cultural experts.

The CQEES could be used in self-awareness training programs. First, from a practical perspective, the system is able to evaluate users and provide them specific cultural training. Second, this system is open in the sense that it can provide a standard interface that can facilitate further development. Third, the CQEES is also extensible, both in terms of the system computational model and the system implementation. Fourth, this system has the potential to work as a training extension agent in order to integrate it into another existing intelligent system.

7 Conclusion

Our research is a pioneer in this field of study as it captures the essence of culture and addresses culture from the intelligence perspective of an individual wanting to develop his ability to adapt to various cultures. In the computer science domain, we have managed to computerize the underlying principles of CQ in order to help individuals improve their ability to adapt to a new culture. CQEES is a system that is "aware of cultural intelligence" and represents a breakthrough with the contributions it makes to both the CQ and Computer Science fields. CQEES serves as an efficient team of top CQ experts that are constantly with individuals or organizations who want to have insights on how to increase their efficiency in culturally diverse settings. Moreover, during the evaluation, the system uses natural language to communicate with users in order to provide the users a stress-free and friendly evaluation environment. The system allows for a better performance than that of human cultural experts and makes it easier for people to improve their cultural skills. In addition, it allows researchers to simplify competitive and complex tasks which are normally carried out manually in the CQ evaluation process. Furthermore, a CQEES based on two AI technologies, can also be applied to other research domains where modeling is useful, such as languages, psychology and sociology.

8 References

[1] Ang, S., Van Dyne, L. “Handbook of Cultural Intelligence”. 1st ed. M.E. Sharpe.Armonk. 2010.

[2] Earley, P.C., Ang, S. “Cultural intelligence: Individual interactions across cultures”. Stanford, CA: Stanford University Press 2003.

[3] Earley, P.C., Mosakowski, E. “Cultural intelligence”. Harvard Business Review, 82, pp.139–146. 2004.

[4] Peterson, B. “Cultural intelligence: A guide to working with people from other cultures”. Yarmouth, ME: Intercultural Press (2004).

[5] Hofstede, G. H. “Cultures and Organizations: Software of the Mind”. McGraw-Hill, New York (1991).

[6] Ang, S., Van Dyne, L. “Conceptualization of cultural intelligence, 2008.

[7] Kon, C., Damien, J., Ang, S. “Cultural intelligence and the global information technology workforce”, NanYang Technological University, Singapore, 2010.

[8] Ang, S., Van Dyne.L., Koh, S.K. “Personality correlates of the four-factor model of cultural intelligence”. Group and Organization Management, 31, pp.100-123, 2006.

[9] Van Dyne, L., Ang, S., Koh, C. “Development and Validation of the CQS: The cultural intelligence scale”. Handbook of Cultural Intelligence. 1st ed. M.E. Sharpe.Armonk, 2008.

[10] Mamdani, E.H., Assilian, S. “An experiment in linguistic synthesis with a fuzzy logic controller”, International Journal of Man–Machine Studies, 1975.

[11] Jang, J.S.R., Sun, C.T., and Mizutani, E. “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”. Prentice Hall, Englewood Cliffs, NJ. 1997.

SESSION

LEARNING MODELS, METHODOLOGIES, TOOLS AND CASE STUDIES

Chair(s)

TBA

Automated Feedback for Personalized Learning

David J. Coe, Ronald Bowman, and Jason Winningham
Department of Electrical and Computer Engineering
The University of Alabama in Huntsville, Huntsville, Alabama, USA

Abstract - *Presented here is an automated grading framework for text interface data structures programming assignments. This framework provides rapid feedback to students, consistency in marking of assignments, and requires minimal time to set up and use. A test driver processes test commands read from input files allowing the framework to support systematic, thorough functional and structural testing of student submissions. The framework generates individualized grade reports summarizing test results and a .csv file that summarizes student grades to speed entry of the grades into our Learning Management System. The automated grading framework has been enhanced to include screening for memory leaks, a common error for students learning to implement container classes in C++. A grade preview mechanism has been derived from the framework to give students personalized feedback on specific defects in their code prior to the final submission deadline, allowing students to prioritize debugging efforts on the most critical functionality.*

Keywords: Personalized learning, structured output assignment, automated program grading, robo-grader, grade preview

1 Motivation

The automated programming assignment grading framework presented here emerged from my experiences teaching introductory computer programming courses in the Department of Electrical and Computer Engineering at the University of Alabama in Huntsville. A driving force behind the development of this framework was a change in the electrical engineering degree requirements. Prior to this change, only our computer engineering students were required to complete the second course of our two-course introductory programming sequence based on the texts by Dale and Weems (see CPE 112 and CPE 212 in Table 1 below) [1-2]. After the program change, electrical engineering students were required to take CPE 212 in addition to CPE 112, doubling CPE 212 enrollments without a corresponding increase in already scarce resources such as graduate teaching assistants. An immediate need for some form of automated grading scheme had emerged.

The automated grading framework presented below was thus developed to satisfy three basic objectives: (1) to provide timely graded feedback to large numbers of students, (2) to make marking of assignments consistent regardless of who was assigned the task of grading, and (3) to maximize the amount of time teaching assistants and instructors had to answer students questions.

2 Systematic Testing Requirement

Given the increasing overall complexity of the software assignments in CPE 212 as compared to those in CPE 112, a personal goal of the instructor was to convey to the students the importance of efficient but thorough software testing. Lectures dedicated to software testing introduced the concepts of functional and structural testing, including examples of how to use the *gcov* tool to identify untested lines of code. Despite these efforts, the quality of code submissions from the typical CPE 212 student was often insufficient to pass instructor-developed tests. In the worst cases, student submissions would compile but fail on every input file used by the instructor, frustrating both the student and the instructor.

Informal surveying of enrolled students suggested that those students who could benefit the most by utilizing *gcov* never tried to use the tool to improve their test set selection, despite the lecture time consumed illustrating its use. So, in addition to the three basic objectives of rapid feedback, consistent marking, and minimal time for setup and use, the automated grading framework should provide a means of demonstrating a systemic approach to testing the container classes.

To explore another possible source of poor student performance, that is, the students did not fully understand all of the project requirements, the instructor produced detailed handouts describing each method and attribute along with sample inputs and outputs illustrating both the desired functionality and desired error handling. It was clear by the questions asked by students during office hours, that many students never read the project handouts. Detailed written descriptions of project specifications were replaced by supplying students direct access to an executable sample solution that would serve as the primary description of the desired product's specifications along with a brief handout describing implementation hints, submission instructions, and assignment constraints. By replacing the written enumeration of requirements with an executable sample solution, the instructor sought to encourage students to explore the desired functionality by developing their own tests for the sample solution that they might subsequently apply to their own code. Again, weaker students struggled as much with developing tests for others code as they did for their own code.

CPE 112 Intro to Computer Programming in Engineering	CPE 212 Fundamentals of Software Engineering
C++ syntax and semantics	pointers
data types	classes
expressions	inheritance
input-output	exceptions
selection statements	polymorphism
looping statements	abstract data types
void functions	sequence containers (stacks, queues, and lists)
value-returning functions	binary trees (heaps and binary search trees)
parameters and arguments	recursion
structures	graphs
arrays	generic programming
top-down design	searching
	sorting

Table 1 – Distribution of topics between CPE 112 and CPE 212

3 Structured Output Assignments

The key element that accelerated the development and deployment of this automated grading framework was the decision to use *structured output assignments*. A structured output assignment can be created when the instructor provides the test driver source code that includes all output statements allowed in the assignment. Students were allowed and encouraged to look at the test driver code, but they were also cautioned to make no modifications to the test driver, or any other source code files supplied by the instructor, since clean copies of those source files would be used when compiling and grading their submission.

The test driver serves as a client of the container class under development and, in some cases, as a client of other classes developed for the assignment. Rather than hardcoding all tests directly within the test driver code itself, the test driver exercises student code by reading the desired sequences of methods to be invoked from a set of input files provided to the students. Each input file contains a series of commands that prompts the test driver to invoke class member functions in a specific order, supplying any required data, and documenting the results by writing to stdout for later capture. Prior to executing any method of the class, the test driver writes text to stdout to indicate the next class method to be executed. This allows students to compare their own outputs to the sequence of requested operations in the input file to identify the point of failure.

Table 2 below shows a representative set of input file symbols and their mapping to corresponding Stack ADT methods. Figure 1 shows a sample use of these symbols to create an input file. Test coverage analysis is performed on the sample solution code to ensure that the set of input files developed will provide rigorous testing. With relatively minor modifications, the test driver and input files can be repurposed for other sequence containers such as queues and lists or even tree structures such as binary search trees. Again, this helps to minimize set up time, making more time available for fielding student questions.

The test driver must also be robust to account for the breadth of errors that may occur. For example, if the *Pop* method for the Stack class is required to throw the *StackEmpty* exception if the Stack object is already empty at the time *Pop* is invoked, then the test driver must trap and document that the *StackEmpty* exception was successfully thrown or not thrown before it continues processing test commands from the input file. Unanticipated exceptions may also occur during execution of student code as a result of segmentation faults, divide by zero errors, and attempts to dereference null pointers. As before, the test driver must trap and document to the desired level of detail any spurious exceptions before it attempts to complete processing of the remaining test commands. For the automated grading framework discussed here, the test driver file provided to the students with each assignment makes extensive use of the C++ *try-catch* construct.

Adjustments to the test driver and input files can also allow students to practice developing both the container code and their own client code that utilizes the their container. In this situation, a subset of the input files will exercise just the container code, ignoring all student client code that may still be in the form of function stubs created by the students for compilation. The remaining subset of input files focus on exercising the client code. So, students see both unit testing and integration testing in action.

Symbol	Description of Test Driver Response	Method Triggered
#	Test file comment	**None.** **Echo comment to stdout**
c	Invoke Stack ADT constructor	**Stack( )**
d	Invoke destructor	**~Stack( )**
p	Print stack object contents to **stdout** (Note: Provided by instructor in **stack.h**)	**Print( )**
+ **x**	Push item **x** read from input file onto stack	**Push(x)**
-	Pop top item from stack	**Pop( )**

Table 2 – Sample input file symbols and their corresponding Stack methods.

```
# p01input1.txt – Sample tests for Stack ADT
c
+ 5
+ 2
p
-
p
d
```

Figure 1 – Sample test driver input file for a Stack ADT that creates a stack object, pushes two integers onto the stack, and then prints the contents of the stack before and after popping the topmost value. Subsequently the stack object is deallocated.

The critical advantage of this approach is that the structured nature of the output greatly simplifies evaluation of the correctness of container operation. Since instructor provided test driver code generates all assignment outputs, elaborate output parsing to account for variations in spelling, capitalization, whitespace, output sequencing, and the inclusion of residual debugging outputs added by the students is eliminated, greatly reducing development time. While the use of structured output assignments is restrictive, the primary focus of the second course is learning the new concepts required to implement container classes in C++ under the assumption that students have achieved some minimal proficiency in writing information to *stdout*. What follows below is a discussion of how structured output assignments, test drivers, and input files are utilized within the Automated Grading Framework.

4 Automated Grading Framework

The automated grading framework consists of a series of four Linux shell scripts that when executed sequentially (1) extracts and organizes files, (2) compiles student submissions, (3) generates customized grading reports for each student, and (4) optionally emails the relevant grading report to each student. While the exact details of these scripts will vary based upon the target platform and Learning Management System used, included below is an outline of the functionality included in the Extract, Compile, Grade, and Email scripts.

For those seeking to replicate this framework, a word of caution – always use a dummy account with minimal execution privileges for grading student work in case a submission contains malicious code – a virtual machine may be of use to reduce the risk of data loss.

4.1 Extract Script

The *extract script's* primary function is to organize all of the required files and set appropriate files permissions. Figure 2 below provides an overview of tasks performed by the extraction script. First, the script creates the directory structure with subdirectories for the sample solution source code, the instructor provided test files, student submissions, and separate subdirectories for each possible grading outcome (no-compiles, score of 0, score of 1, etc.). With the directory organization in place, source code for the sample solution and all instructor-provided input files are moved into their respective subdirectories, and read-only permissions are assigned to files that should not change as submissions are graded. Next, the contents of the compressed file of student submissions from Angel, our Learning Management System (LMS), are extracted into the *previous-submissions* subdirectory.

- Create directory structure
- Move instructor-provided source code and test files into respective subdirectories
- Set read-only permissions on key files
- Extract submissions archive file from LMS into *previous-submissions* directory
- Adjust submission directory names assigned by LMS
- For each student, move most recent submission to *submissions* directory

Figure 2 – Outline of *extract script* used in this Automated Grading Framework

- Build sample solution
- For each student submission
    - Use *dos2unix* utility to strip incompatible characters from student files
    - Add instructor-provided *Makefile* and source files to student directory
    - Compile submission using the *make* utility and log build issues to *build.txt*
    - If build fails, move student subdirectory to no-compile directory

Figure 3 – Outline of *compile script* used in this Automated Grading Framework

The Angel LMS includes copies of all submissions a student has submitted to the online assignment drop box – even if the last submission was tagged for grading within Angel. So, if student X submitted a preliminary version of their code prior to their final code submission, then both submissions will appear for student X within the compressed file. For convenience, student submission subdirectories are renamed with the student's username prepended to the directory name. The most recent submission from each student is identified and moved into the *submissions* directory leaving all older submissions within the *previous-submissions* directory that will not be graded under the assumption that the most complete and correct submission will be the last submission made by the student.

4.2 Compile Script

The *compile script's* primary responsibility is to compile the sample solution source code and each student submission. If a student submission does not compile, the student's subdirectory is relocated to the no-compile directory. Since many students utilize computers running the Windows operating system, it is important to use the *dos2unix* utility to remove any characters that will impede compilation. Figure 3 above is a step-by-step outline of the compilation script.

4.3 Grade Script

The *grade script* scores each assignment and sorts them by score received into separate subdirectories. Each test file that results in correct outputs with no memory leaks counts as one point towards the final score for the assignment. The grading script uses the Linux utility *sdiff* to verify that the functional behavior of a student submission matches that of the sample solution, i.e. the detailed project specification, and any behavior difference identified is treated as a failure. The leak check feature of the *valgrind* utility is used to identify any memory leaks encountered during processing of a particular input file. Any memory leaks encountered are also treated as a failure for that input file. The results of these analyses are documented in a *grade.txt* file that serves as the summary that will be returned to the student. A more detailed outline of the grade script appears in Figure 4 below.

It is important to note that the *ulimit* utility restricts resources allocated for the execution of the submission to contain incorrect behaviors, such as infinite loops, that are not addressed by the exception handling of the test driver code. The exit status may be used to gain additional insight regarding unexpected termination of the student code. After each assignment grade is finalized in the *grade.txt* file, an entry is appended to the assignment *gradebook.csv* file that contains the student's username and assignment score. Importing this file directly into the Angel LMS places all student scores into the online course grade book with no manual entry required.

4.4 Email Script

The optional *email script* may be used to forward each student's grade.txt report to the student's campus email address.

5 Grade Preview

The *preview script* is a modified version of the *grade script* that allows students on demand to see a preview of the grade their code would receive if it were submitted in its current state. The preview script points out specific anomalous results that would cause a student to loose points if left uncorrected. Prior to the preview script in Fall 2008, the average score on the queue-based assignment was 76.5%. In Spring 2014, the average score on the corresponding assignment increased to 84.8%. This automated personalized feedback is also accessible to students even when instructors and teaching assistants are unavailable. The preview script has also proven invaluable when the instructors and teaching assistants are assisting students since it provides a quick summary of issues.

```
•  For each test file
      o  Execute sample solution saving output into a text file

•  For each student whose program compiled
      o  Create student's grade.txt and append student identifier
      o  Append build log contents to grade.txt
      o  Set score = 0
      o  For each test file
            ▪  Execute student submission saving output in text file using ulimit on execution time, file sizes, etc.
            ▪  If exit status not successful,
                  • Append appropriate description of result to grade.txt file for that input file (segmentation fault,
                     time exceeded, etc.)
                  • Continue to the next test file
            ▪  Otherwise,
                  • Verify student outputs using sdiff to compare student outputs to sample solution outputs &
                     append results to grade.txt file.
                  • Verify no memory leaks by using valgrind & append memory leak analysis results to grade.txt
                  • If outputs match exactly and no memory leaks then increment score by one for that input file.
      o  Append overall score to grade.txt
      o  Append username and grade to gradebook csv file
      o  Echo grade.txt to stdout
      o  Update summary histogram variables

•  For each student whose program did not compile
      o  Begin writing grade.txt file by adding student identifier
      o  Append build log build.txt contents to grade.txt
      o  Append grade of zero to grade.txt
      o  Append username and grade to gradebook csv file
      o  Update summary histogram variable

•  Write summary statistics for assignment to stdout
      o  Frequency of occurrence of each score, including number of no compiles
      o  Total number of students submitting work
      o  Average score on the assignment
```

Figure 4 – Outline of *grade script* used in this Automated Grading Framework

```
•  For each student whose program compiled
      o  Email grade.txt to that student
•  For each student whose program did not compile
      o  Email grade.txt to that student
```

Figure 5 – Outline of *email script* used in this Automated Grading Framework

6 Conclusions and Future Work

The Automated Grading Framework presented here has a number of advantages. Since it requires minimal time to set up and use, it can provide rapid feedback to students after the assignment submission deadline, even when grading dozens of submissions. This rapid feedback gives students the opportunity to learn from previous errors in time to avoid repeating them on the next assignment. Moreover, this framework sets a consistently high marking standard and provides uniform marking results regardless of which instructor or teaching assistant executed the scripts. From the student perspective, by providing round-the-clock access to individualized feedback, the *preview* script helps students to continue making progress regardless of when and where they choose to work on the assignment.

Despite these advantages, there are disadvantages to using this framework. The structured nature of the assignments may lead to more similarities in student code submissions making plagiarism detection more difficult. The instructor currently compensates for this possibility by reducing the total course credit for projects, by screening submissions using the MOSS plagiarism detection tool to identify potentially plagiarized submissions for manual screening [3], and by asking assignment-derived questions on the exams.

Another disadvantage is that in the end, the goal is have the students learn how to test their own programs.

The test driver source code and test files provide an example of a systematic approach to testing of the container, but the instructor currently supplies these materials. Future work will investigate the possibility of teaching about test coverage analysis by using this framework backwards – that is, supplying the test driver and the code under test and requiring students to develop and submit sets of test files as a graded assignment, with *gcov* analysis used to assess the completeness of the test set.

7 References

[1] Nell Dale and Chip Weems, *Programming And Problem Solving With C++, 6th editon*, Jones & Bartlett Learning, March 6, 2013.

[2] Nell Dale, *C++ Plus Data Structures, 5th edition*, Jones & Bartlett Learning, September 26, 2011.

[3] Alex Aiken, *MOSS: A System for Detecting Software Plagiarism*, URL http://theory.stanford.edu/~aiken/moss/

A Case Study on Introducing the Design Thinking into PBL

Shuhei Ohsako[1], Yasutaka Kamei[1], Shintaro Hosoai[1], Weiqiang Kong[1], Kimitaka Kato[2], Akihiko Ishizuka[3], Kazutoshi Sakaguchi[3], Miyuki Kawataka[4], Yoshitsugu Morita[5], Naoyasu Ubayashi[1], Akira Fukuda[1]
[1]QUTE: Kyushu University Research Center for Advanced Information and Communication Technology Education, Kyusyu University, Fukuoka, Japan
[2]Fujitsu, Ltd., Kanagawa, Japan
[3]FUJITSU DESIGN, Ltd., Kanagawa, Japan
[4]FUJITSU UNIVERSITY, Ltd., Kanagawa, Japan
[5]Graduate School and Faculty of Design, Kyushu University, Fukuoka, Japan

Abstract - *In this paper, we describe a case study on introducing the Design Thinking as preparations for Project Based Learning (PBL). The purpose of introducing the Design Thinking is to improve students' skill in producing new added-value at the time of solving project problems and investigating PBL themes. In addition, we describe the results of the questionnaire carried out for evaluating the effect of this introduction.*

Keywords: Design Thinking, Project Based Learning

1 Introduction

The curriculum of Kyushu University Information Communication Technology Architect Educational Program (QITO) [1], which is offered in cooperation with companies like Center for Future ICT Leaders (CeFIL) [2], has three educational approaches: "PBL (Project Based Learning)"[3], "ICT All-Round Education"[4], and "Long-Term Internship". These approaches are adopted for dealing with insufficiency in conventional education. In this paper, we focus on the PBL approach.

QITO aims at upbringing the pi-type engineers, as illustrated in Figure1, who are going to lead the next-generation information-oriented society. In particular, we aim for upbringing engineers who have "Design Ability" that connects "Technology Ability", the knowledge of computer science, and "Creative Ability", the utilization of ICT for profiting society. QITO introduces value-creation oriented PBL to raise "Design Ability".

In value-creation oriented PBL, students will learn "Technology Ability" (development skill, development technique, etc.) and "Creative Ability" (problem discovery and solution, etc.), and additionally, students are expected to design new added–value that raises society innovation, when they examine the theme of the PBL projects and the solutions of the problems.

To this end, we introduce a lecture and the practice of the Design Thinking before the start of value-creation oriented PBL, so as for the students to learn techniques to create new services and ideas, and to raise "Design Ability" necessary for a pi-type engineer.

In this paper, we describe the contents of the Design Thinking lecture that we carried out in PBL, its position in the PBL, and the effect of Design Thinking introduction. In section 2, we describe a summary of value-creation oriented PBL introduced in QITO. In Section 3, we describe the contents of the lecture and the practice of the Design Thinking that we carried out. In Section 4, we describe the effect of the Design Thinking introduction based on the results of questionnaires conducted after the lecture and the comments to the achievements of the practice. Finally, in Section 5, we summarize this paper and describe future prospects.

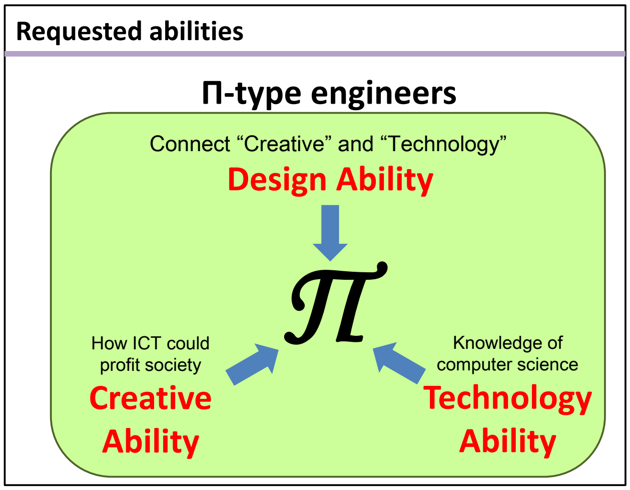

Requested abilities
Π-type engineers
Connect "Creative" and "Technology"
Design Ability
π
How ICT could profit society
Creative Ability
Knowledge of computer science
Technology Ability

Figure1 Desired abilities for pi-type engineers

2 Project Based Learning

2.1 Summary

In the Fundamental PBL carried out in the first term of first-year master course, students are given a common problem of expanding an existing Web system for the purpose of acquiring basic software development and project management skills.

On the other hand, in the Practical PBL carried out in the second term of first-year master course and in the Advanced PBL carried out in first term of second-year master course (Practical and Advanced PBL are treated as value-creation oriented PBL), each student team start-ups a project on a different theme, where these themes could be society problems provided by teachers or companies, or students self-planned themes, etc. In addition to techniques necessary for completing a project theme, another purpose of such practice is to let students acquire abilities to discover problems and to think about means to solve the problems by students themselves based on teamwork.

2.2 Value–creation oriented PBL

In value-creation oriented PBL, each team works on a different theme as described follows. Because solutions are not prepared for these themes in advance, it is important that students could come up with solutions with higher added value when they solve the problems of the project.

- System development type (Company collaboration type)
  The purpose of this theme type is to develop systems according to requirement specifications provided by companies or teachers.

- Contest challenge type
  This project aims at winning the championship at ESS robot challenge [4]. Through using a vacuum cleaner type robot, students examined and developed algorithms for making the map of the competition field, as well as for arriving at the destination within specified time.

- International type / research type
  The purpose of this theme is to conduct research, in collaboration with an overseas university, on topics requested by teachers.

- Business planning type (Entrepreneurship)
  Students propose services and systems, which have the potentials to be developed into new business, and conduct the actual development. One example of past themes is "plan and development of a Web Service to visualize topic flow"[6], which was proposed and developed by students themselves in 2013.

3 Design Thinking

3.1 Summary

The Design Thinking is a technique to create new service and business causing innovation to bring the society a change[7]. The approach called "Co-creation" is important in uncertainty and diversifying society. "Co-creation" is to adopt the wisdom from every person like businessmen and customers as well as designers, and to create services.

The first step of service creation in the Design Thinking is to notice exiting problems by inspecting a field. Then, a prototype and a business model of the expected service are created by considering solutions to the problems by teamwork. Furthermore, the service and business model are brought into a close on-site examination in repeated cycles, during which new problems are extracted through inspecting the field by using the prototype developed.

3.2 Aims of the introduction

We introduced the Design Thinking lecture so as to let students learn how to solve project problems and to come up with more attractive ideas, which are necessary for value-creation oriented PBL.
Aim 1: Let students learn techniques and means of thinking to create new ideas for solving society problems, and to learn the process of reaching mutual agreement.
Aim 2: Let students deepen their understanding about the importance of "The Three Actuals" (actual place, actual part, and actual situation) for solving problems, rather than learning impractical theory.

3.3 Schedule

We carry out the lecture before the start of the Practical PBL in the second term of first year master course to utilize the Design Thinking when students examine the theme of business planning type and create ideas in PBL. In 2013, we carried out the Design Thinking lecture as part of a lecture called "Advanced Topics on Social Information Systems" to demonstrate examples of the social problem solutions and innovation essence using ICT.

The lecture of the Design Thinking was carried out in group work on July 4 and July 11, and an achievement presentation was conducted on July 19. The aim is to create new ICT services by using the Design Thinking, based on the lectures about the utilization of the ICT in real scenarios of social infrastructures (medical care and agriculture, etc.), which were carried out nine times from May 9 to June 20.

3.4 Practice contents

We set a virtual project in this lecture for the purpose of "Raising social innovation in Kyushu University Library".

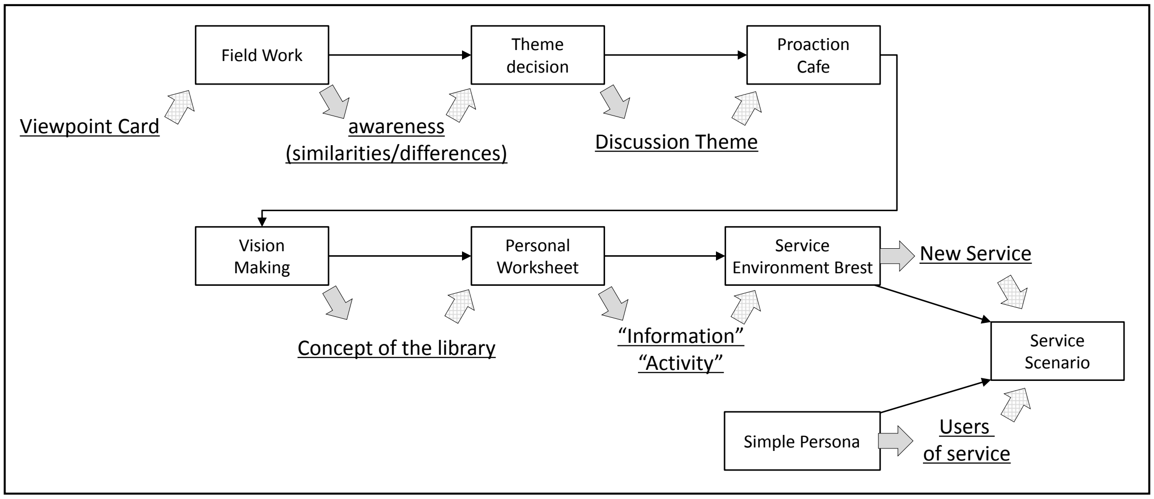


Figure2 Work flow and works in the practice

The time schedule of the group work is shown in Table1. We chose a library as the theme of the practice, because students have a well-understanding of it and can perform the field work easily when they investigate new services. The work flow and achievements in this practice are shown in Figure2. It was intended that students just create new innovative services in this practice without necessity for examining the feasibility and possibility of commercializing the services. In the Design Thinking, it is important to think and discuss with a positive attitude. Therefore, we set the following rules when we carried out the practice.

- To talk frankly about personal thinking and feeling without hesitating.
- To accept comments from other persons with a positive attitude, without negation and judgment.
- To enjoy a discussion with many scribbling memos on tags.

Table1 Time schedule of the group work

Day	Time	Contents	Unit
Jul 4	50min	Field work	Pair
	20min	Share of awareness	Pair
	10min	Theme decision	Whole
	20min	Pro-action cafe	Group
	30min	Vision making	Group
Home work	-	Personal worksheet	Individual
Jul 11	60min	Service environment brest	Group
	60min	Simple persona	Group
	60min	Service scenario	Group

3.5 Practice system

People who constitute this practice are shown in Figure3. We invited lectures from Fujitsu and Fujitsu Design Limited because they have experience in examining new services in actual fields. In addition, as an important part in the Design Thinking, we carried out information gathering and problem hearing of the actual field in the Ito Library of Kyushu University. Furthermore, students of Kyushu University Graduate School of Design joined the practice as facilitator who made the discussions more activate and supported to visualize proposed ideas.

3.6 Practice contents of the first day (July 4)

3.6.1 Field work

Two students make a pair and perform the field work in the Ito Library of Kyushu University so as to understand the present conditions of the library. The explanation tour guided by the library staffs is carried out in the library and students receive the explanation of the functions of the library. Next, every pair observes the library and performs a hearing of the library staffs and users.

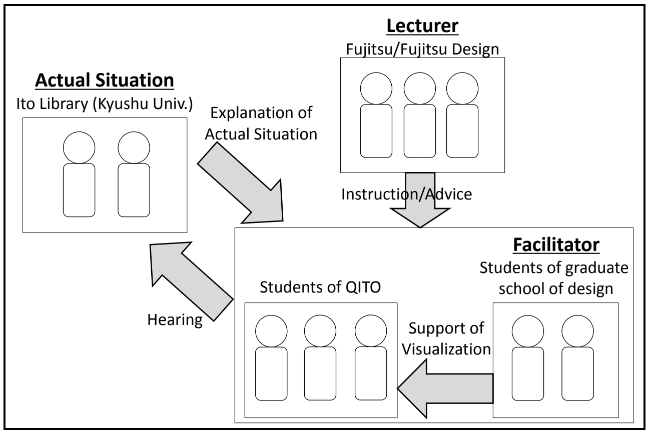


Figure3 Lecture system in this practice

Table2 Discussion theme and concepts of new libraries created in this practice

Viewpoint card	Kind	Theme (Keyword)	Concept
Smoking area	Similarities	There are persons who do not use it at all	Enjoyable leaning
The Apple store	Similarities	There is Macintosh	Appealing that something is there
Video arcade	Similarities	The characters of the users vary according to floors	Height of floor shows height of will
Sushi shop	Differences	There is home delivery	Handbill operation
Airport	Similarities	It is used other than its original purpose	Library to feel

We use the "Viewpoint card" as means to get awareness of the library. A name of the public facility, such as an exhibition, a hospital, a bar, or a restaurant (except the library) is written in a viewpoint card. The viewpoint card is distributed to each pair at random beforehand. When each pair performs the field work in the library, they carry out inspection and hearing in a viewpoint of similarities/ differences between the library and the public facility written in the viewpoint card, and they take notes of the awareness about the characteristics and the function of the library.

3.6.2 Share of awareness / Theme decision

Each pair shares the awareness gotten in the library. Each pair shares the contents of the memo written in the card, and writes keyword on tags in a viewpoint of similarities/differences between the library and the viewpoint card, and puts it on the white board.

Then, further discussion themes are chosen from the keywords extracted by each pair. Each student votes for five keywords that he/she consider important from all the keywords extracted by other pairs. After that, keywords with the most votes are selected as discussion themes. In this lecture, five themes shown in table 3 are selected as discussion themes. For example, library users may use study hall and learning consultation in addition to the original borrowing books purpose; airport users may not only board an airplane but also buy souvenirs and eat meals. Therefore, the keyword "It is used other than its original purpose" was extracted, and selected as a discussion theme.

3.6.3 Pro-action cafe

While digging into the extracted themes, students construct groups to examine ideas for each of the new library themes. At first, each table is assigned a discussion theme, and one student is remained at last as a host for the theme. Then, other students who have interests in this theme may move to the table and further discuss about the theme. Students except the host may freely move to other tables/themes in the middle of the discussions. Students discuss themes at each table, and perform development and deepening of the themes. Japanese vellum is prepared at each table, and students write down new opinions extracted from the discussions. After having repeated movement several times, students choose the theme that they sympathize most, and form groups for the following practice.

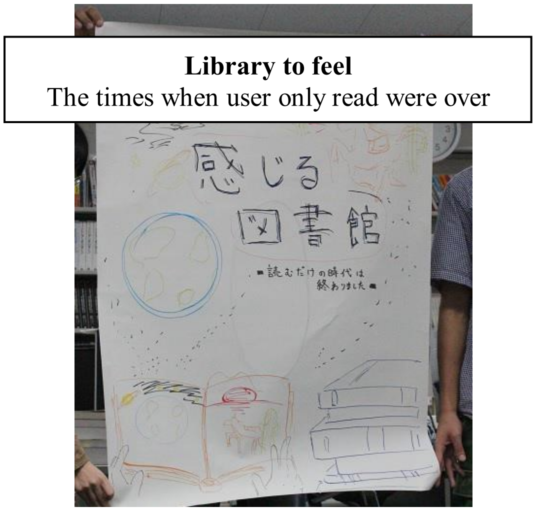
3.6.4 Vision making

Each group investigates the concept of the selecting theme, and makes the cover catalogue of a new library. It is important that each group thinks about how they attractively convey the appealing point of the new library by using not only wording but also illustration. Concepts made by this practice are shown in Table2. For example, as shown in Figure4, as a result of digging into the theme "used other than its original purpose", a concept of "Library to feel" was made for the purpose of acquiring new constituencies through installing the theme parks, by which users can experience a world view of books.

3.6.5 Personal worksheet

As homework to be done before the practice of the second day, each student writes down on a worksheet his/her viewpoints of the theme, as the starting point of his/her ideas. By imaging an ideal operation state of the new library according to each theme, students summarize according to the following items.

- All "participants" who use the new library
- "The exchanges" that are carried out in the new library



Library to feel
The times when user only read were over

Figure4 Examples of the concepts of a new library

- "Feeling" of the participants who use the new library
- Solved small "inconvenience" and "social problems"
- "The value" that is brought about in the new library

3.7 Practice contents of the second day (July 11)

3.7.1 Service environment brest

The worksheet contents of each student are shared within the group. From the worksheets, each group extracts keywords about the "information" concerning the library (books, users, floors, etc.), and about the "activity" which expresses the action of users, and then record these two information on tags. For example, as "information", keywords about users (such as "couple", "parent and child"), and about the characteristics of the library (such as "local oasis", "bad access", and "silent") were extracted. As "activity", keywords such as "search", "study", as well as "eat and drink" were extracted.

After that, each group thinks about the ideas of the new services by combining the "information" and "activity" extracted. Then, one idea should be summarized in one piece of A4 sheet. It is important that students think about combinations as much as possible within specified time. The quantity, rather than the quality, of the ideas are attached more importance.

3.7.2 Simple persona

To concretize the usage scenario of the proposed service ideas, simple personas of different usage styles are made. For each persona, a concrete profile such as hometown, hobby, character, and sense of values is set. In addition, each group thinks about how a persona uses a service idea in a chronological order. The image of each persona should be summarized in one piece of A4 sheet by using not only words but also illustrations.

3.7.3 Service Scenario

Each group realizes services by considering concrete scenarios through combining service ideas and personas, and summarizes the design of the new library. Each group builds up concrete service scenarios with sympathy, while using illustration on one piece of sheet by putting together the combination of "information" and "activity". Such combination becomes the starting point of the service, the use scene, the scenario that related with the scene, and the persona concerning with the scene.

One example of the service in the "Library to feel" is shown in Figure5. The service is to start a restaurant in the library that serves foods. It is expected that persons, who don't usually read books and are only interested in eating, may come to visit the library for using the restaurant, and then may become interested in books in which dishes offered by the restaurant appear.

3.8 Achievement presentation (July 19)

Each group presents its design, and conveys the concept of a new library to participants. In this practice, the presentation documents were made by utilizing illustrations and images, and the presentation time was set to 25 minutes per group, including question and answer time.

Each group presented the concrete suggested service scenarios. The presentation explains how a "persona" (which was set with concrete profile) used the "service" (which took into account the combination of "information" and "activity") in the new concept library. Examples of suggested services by each group are shown in Table3.

Students' fieldwork reveals that there were few users of the Ito library. To deal with this fact, students proposed services for increasing user numbers of the library. The service category is as follows.

- Enhancement of students' will to use the library

For the purpose of raising students' will of using the library, new services are proposed, which include, e.g., visualizing students' efforts and providing better services according to usage frequency; and recommending research laboratories according to book rental history.

- Expansion of the constituencies of the library

To attract users who do not usually use the library, it is proposed to set up theme parks and hold events such as quiz contest or lottery. It is expected that such means can attract new non-student users like children and housewives

- New encounter / interchange

Through setting up places for task display or hobbies, and photograph galleries, the proposed services are expected to be able to deepen interchange among not only students who use the library but also local inhabitants.

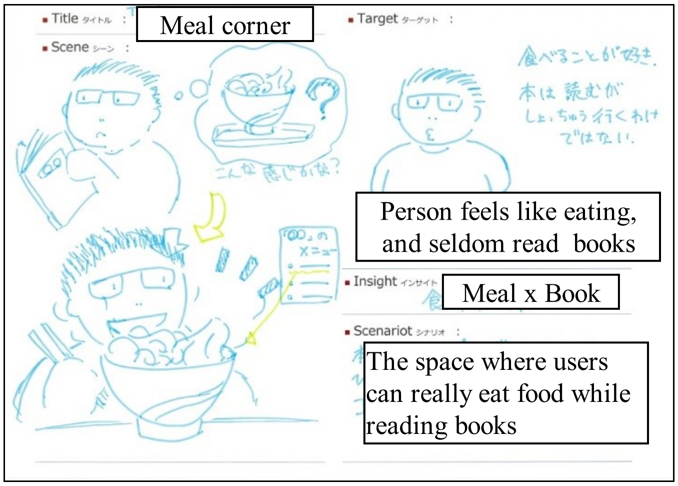


Figure5 Example of a service scenario provided in a new library

Table3 Examples of suggested themes provided in a new library

Concept	Information	Activity	Persona	Contents of service
Enjoyable leaning	indifference	go to library	family, child	holding a contest to look for the answer to a problem from books
	grandchild generation	use PC and mobile	child, old person	promoting local interchange through teaching the operation methods of PCs and smartphones by a child
Appealing that something is there	everyone	search books	foreign student	searching and displaying the route to a book using tablet terminal
	advanced	somehow	student of varied hobbies	increasing friends with a common hobby by setting up places for free talks
Height of floor shows height of will	books	public lottery	students, housewives	performing a lottery using the ISBN of books among people who borrowed the books in the past
	library	use facilities	students, professors	making comfortable rooms available for people who uses the library more times
Handbill operation	teachers, students	rental history	Students	recommending a suitable laboratory to a user based on his/her book rental history
Library to feel	physical science	introduce products	student for the encounter	providing places for persons who have interests in the same fields through displaying student themes
	infants	experience	mother with children	not only making comfortable usage available for housewives, but also attracting children's interests to the library

The ideas created by this practice received the following comments from lecturers and participants.

- More library services are concretized than expected, and the services are considered to be feasible with respect to realization. However, since the fieldwork was conducted only in a short time, it is necessary to further dig into demands from actual situations.
- Because it has been shown that problem solving abilities are improved by repeating the cycle of idea creation, such techniques learned from this practice should be continuously utilized in the future.
- The created ideas were slightly felt insufficient from the viewpoint of innovation. At the initial stage of idea creation, it is expected that students could catch more the operation mechanism of the library.

4 Effect of introducing the Design Thinking

4.1 Summary

We carried out a questionnaire to students after the achievement presentation with the purpose to evaluate the effect of introducing the Design thinking and hence to improve the lecture contents for the next fiscal year. The questionnaire is in a free description form and students can describe their impression and thought experienced during the Design Thinking.

4.2 Effect

The results of the questionnaire are shown in Table4. About thinking techniques, because the students could realize the effect of introducing the Design Thinking, we obtained the result that there was constant effect for future PBL practice. About the structure and process of the practice, there was the opinion that argument was activated by students from Kyushu University Graduate School of Design, who joined the practice as facilitators. On the other hand, there were indications about motivation and time designation of the practice, which are considered as future tasks.

Furthermore, about the contents of the achievement presentation and the process of the lecture, the comments as described below were obtained from the people related to Kyushu University Library.

- The services were very unique and interesting. In the future, students are expected to be able to think about things by innovative approaches like used in this practice, without narrowing a range from the beginning.
- Because students only visited the library at the time of fieldwork of the first day, they should investigate the services already provided by the library and collect demands from actual situations other than pure lecture time.
- There were many public library oriented services, and few ideas targeted at university libraries.

Table4 Questionnaire results about the practice of the Design Thinking

Classification	Good points		Points to be improved	
	Number of responses	Contents	Number of responses	Contents
thinking techniques	17	Because more opinion exchange was conducted than conventional discussions, many new ideas were able to be created.	3	Lectures about the process of putting the ideas into practical use (e.g., whether the ideas are feasible with respects to techniques and cost) are necessary
		Suggested services were able to be concretized by understanding the problems existing in actual situations.		
process of the practice	8	The visualization method and the way of thinking of the students from Kyushu University Graduate School of Design were very useful for reference purposes.	6	It was vague whether ICT should, and to what extent, be utilized in the practice.
				Because there was too much work in the practice, understanding of the importance of each process seemed to be shallow.

5 Conclusion

We introduced the lecture of the Design Thinking as a preparation for value-creation oriented PBL. The purpose is to let students learn the thinking methods and the process of reaching agreement for creating added value when they solve problems, and let students understand the importance of solving problems based on opinions from actual situations, rather than based on impracticable theories.

As the results of the questionnaire after the Design Thinking lecture, we understood the effect of such PBL practice including " Through the methods of the Design Thinking and the way of thinking of students from Kyushu University Graduate School of Design, more ideas could be created than convention discussions" and "the importance of making clear what is happening in actual situations could be understood better".

On the other hand, some issues for improving the lecture contents were pointed out, such as "lectures about the process of putting the ideas into practical use are necessary" and "the investigation of actual situations was short". Furthermore, the refinements about the accomplishment degree of the practice were also pointed out, such as "the ideas for solving the problems of university libraries were insufficient". Therefore, we will feed back the comments for the process of in future lectures.

6 Acknowledgement

This research is carried out in the programs of "Project for Establishing a Nationwide Practical Education Network for IT Human Resources Development (Education Network for Practical Information Technologies)" and "IT Specialist Program Initiative for Reality-based Advanced Learning" supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. Furthermore, this research received the cooperation of the Ito Library (Kyushu University Library) at the time of the field work and the evaluation of the practice results.

7 References

[1] Website of QITO, Kyushu University http://www.qito.kyushu-u.ac.jp/en/

[2] Website of CeFIL (In Japanese): http://www.cefil.jp/

[3] Akira Fukuda, Naoyasu Ubayashi, Keijiro Araki, Tsunenori Mine, Shigeru Kusakabe, Kunihiko Kaneko, Yasutaka Kamei, Norimichi Hiroshige, Zenjiro Ohba, Kaoru Nakatani, and Keizo Tatsumi. *PBL practice guide for university teachers of computer science*, 2012. (In Japanese) Download Website (In Japanese): http://www.qito.kyushu-u.ac.jp/en/publications-presentations/publications/3826

[4] Tetsuya Ohishi, Weiqiang Kong, Norimichi Hiroshige, Naoyasu Ubayashi, and Akira Fukuda. *Remote lecture using many spots connection device*, Research report of JET Conferences 12(4), pp.73-80, 2012. (In Japanese)

[5] Website of ESS Robot Challenge (In Japanese) http://www.qito.kyushu-u.ac.jp/ess/2013/index.html

[6] Branchat (In Japanese): http://branchat.net/

[7] Tim Brown, *Design Thinking*, Harvard Business Review, 2008.

The Impact of Cultural Factors on Technology Acceptance, Students Point of View

FECS'14: The 2014 International Conference on Frontiers in Education: Computer Science and Computer Engineering

Dhiya Al-Jumeily[1], Abir Hussain[1], Sam Crate[1]

[1]*Applied Computing Research Group, Liverpool John Moores University, Liverpool, Merseyside, L3 3AF, UK*

Abstract - *This paper proposes a new technology acceptance model (TAM) to support the development of eLearning system. The focus is thus laid on creating an adaptive eLearning system that factors-in user acceptance based on cultural influences. The TAM proposed in this paper thus includes self-efficacy, system accessibility, subjective norms, perceived ease of use, behaviour intention, and perceived usefulness based on the TAM as proposed by well-known research. An experiment was reported as part of this research that involved one group of students in Oman and one in the UK. This experiment measured the differences in TAM outcomes between students who have grown in Oman and the UK. The challenge members of the Middle Eastern and Gulf Arab communities, is that the internet, ICT and e-learning have arisen and are therefore inevitably associated with western contemporary culture, which is different from Islamic culture.*

Keywords: Technology acceptance module, self-efficacy, system accessibility, subjective norms, perceived usefulness

1 Introduction

Culture is a common behaviour that is learned from the society into which one is born—or in which one becomes embedded—and consists of norms, beliefs and customs [1]. Moreover, it reflects the common set of values that characterize a society in the forms of family, education and the system of social organization [2]. Any particular society or community might accept certain values whilst rejecting others owing to the nature of any particular individual or grouping of individuals, on the one hand, and on the other hand owing to the social, intellectual, religious and political background of the wider society/community [2].

Several studies have emerged over the last thirty years to show that cultural background plays a significantly important role in affecting the uptake and use of technology [3], [4]. In the case of ICT, most information systems have been developed within the Western world and thus are culturally biased in terms of those societies, and even in terms of the mainstream cultures within those societies [5]. Because of this western cultural influence ICT may be presented to non-western societies in forms that are not necessarily appropriate for non-western cultures. Collis [6] makes the argument that culture is a crucial factor that influences how humans accept, use and react to the internet.

2 The Technology Acceptance Model (TAM) and E-Learning

The use of modern technology has long influenced increasing numbers of aspects of people's social and work lives. However, the case of ICT is unique, owing to the phenomenal rapidity with which it has spread around the world and is increasing its presence in the lives of populations on a global scale hitherto unseen. The impacts of this rapidity and extent of pervasiveness have driven much of the research into technology diffusion and acceptance. This in turn has influenced the development of a variety of theoretical frameworks and models that investigate technology acceptance in general and particular, including the works of Davis [7], [8], Davis et al. [9], Hess et al. [10], and Cornell et al.[11].

Davis [7], [8] first formulated the TAM based on the Theory of Reasoned Action (TRA) that had been developed by Fishbein & Ajzen [12] from previous research on the theory of attitude, which led them to the study of attitude and behaviour. Their theory sought to explain behaviour through the observation and analysis of subjects' attitudes, declared intended behaviour and actual behaviour. The separation of behavioural intention from actual behaviour in this theory further enabled them to offer explanations of limiting factors on attitudinal influence and thus to build a model for the prediction of behavioural intentions [13]. The approach adopted by Davis [7], [8] on the basis of TRA theory posits that actual behaviour is associated with the intention(s) underlying a person's behaviour, and he elaborated this into a model for studying and explaining the acceptance of new technology. The TAM has been developed into subsequent versions by Davis and others [9], [14], and [15]. Numerous studies have employed the TAM to investigate specifically the uptake of e-learning, such as those by Park [16] and Arenas-Gaitán et al. [17].

According to Suh & Lee [18], two dominant factors make the TAM ideal for examining the adoption of e-learning systems: users' attitudes to ease of use and perceived usefulness. Lee [19] identified other factors such as perceived adequacy of facilities/resources, internal versus external computing training, internal versus external computing support, and external equipment accessibility. Another factor identified by Suh & Lee [18] is perceived enjoyment, and they showed that perceived enjoyment has an important impact on the intention of using e-learning and on actual e-learning usage. The main purpose of TAM is to predict people's attitudes, behaviours and intentions to new technology as they are formed by external variables as shown in Figure 1.

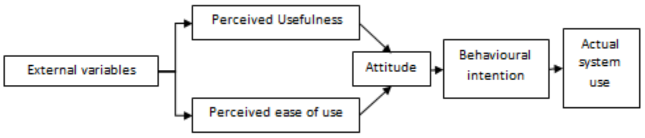


Figure 1: Technology Acceptance Model [TAM] (Davis 1989)

3 Proposed TAM with Cultural Factors

This model comprises the TAM factors identified by Davis [8] together with PU, PEOU, behavioural intentions and attitudes relating to accepting and using new technology. The model also incorporates three main external factors as manifested in (i) social factors, (ii) cultural factors and (iii) political factors, owing to the varied influences exerted by culture on human behaviour. Cultural neutrality has been identified as a blind spot in previous TAMs, because culture has been demonstrated in the literature to exert a major influence on acceptance. Unfortunately, the literature shows that technology is predominantly developed for the young [20]. Often the practice when dealing with nonwestern cultures is to take the existing knowledge regarding technology acceptance in developed western nations and to relate it to other cultures based on cultural beliefs and values [3]. However, as Ziefle & Jakobs [20] affirm, technology users across the globe have different perceptions, styles of thinking, cognitive and cultural values, and assumptions.

Social factors include language, qualifications/skills, and facilitating conditions. The language used in technology plays an important role in a user's attitude towards technology. When technological language is easy and understandable, the use of technology will be easy and flexible, which elicits positive attitudes towards that technology. The converse is also true, as difficult and complicated technological language generates negative attitudes towards technology. Other important social factors include qualifications and the skills required to use technology. Facilitating conditions refer to the technical support available and the adequacy (or otherwise) of equipment and/or software. All these are indicated as important factors by the literature [21], [22]. Three primary continuums drawn from the cultural dimensions theory of Hofstede [23] are used to identify the differences in the cultural factors—individualism/collectivism, uncertainty avoidance, and power distance.

Individualism/Collectivism is the degree to which individuals are integrated within a group and whether individuals are more loyal to themselves and immediate family versus a larger group [24]. Uncertainty Avoidance is defined as the tolerance of a society for uncertainty. High uncertainty-avoidance cultures implement rules and laws to support plans that are followed step-by-step to minimize unknown and ambiguous circumstances while low uncertainty-avoidance cultures have fewer rules and procedures. Power Distance reflects the way people accept and perceive power differences. High power-distance cultures accept autocratic power relationships, the higher the power culture the more power is concentrated in the hands of an elite few [22].

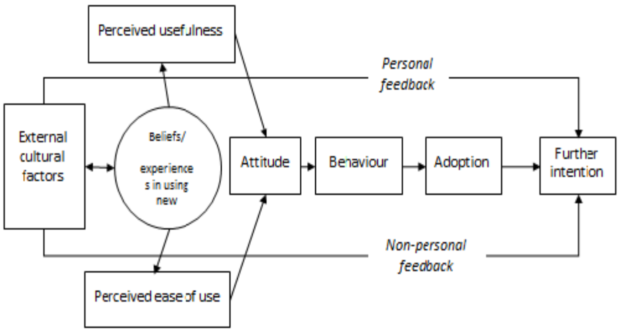


Figure 2: Proposed Technology Acceptance Model (TAM)

The TAM proposed by this paper is shown in Figure 2. In the diagram there are two special flow-lines that designate *personal feedback* and *non-personal feedback.* It might be argued that these are superfluous or redundant in this model. However, it is suggested here that the discussion of the TRA and TAM illustrates the complexities involved in human motivation, especially where *intended* behaviour often gives way to *actual* behaviour, even against a person's better judgement, owing to factors of which the person might not be fully aware. With regard to the discrepancy between intended and actual behaviour, Ajzen [25] first introduced the concept of planned behaviour and subsequently developed it [26], [27] whilst Bandura [28] formulated the concept of *self-efficacy* (referring to an individual's confidence in his/her personal competence) and explored the tensions between this and the individual's *expectation of the outcome* of a course of action [29]. In other words, a person does not always implement his/her planned behaviour, and does not always exercise self-control. In Figure 2 the outlying arrows for *personal* and *non-personal* feedback refer to those influences that might cause an individual to modify intentions and/or behaviour even at the last instant before engaging in a certain course of action or behaviour.

This research thus critically recognizes that the success of using information technology in an e-learning environment will not only have to take cognizance of the individual's level of familiarity with technology, but also take into account that various other factors will have an effect on the learner's experience, perceptions, performance and (ultimately) acceptance of the e-learning process. The focus is thus laid on creating an adaptive e-learning system that factors-in user acceptance based on cultural influences. This is important since cultural influences affect each individual's responses, as well as the level of analysis. As Srite & Karahanna [24] state, cultural values such as "masculinity/femininity, individualism/collectivism, power distance, and uncertainty avoidance are incorporated into an extended model of technology acceptance as moderators". Lau & Woods [30] have identified that the characteristics of a learning object influence "perceived usefulness and perceived ease of use of learning objects, therefore, individual differences appear to have no influence upon intention to use learning objects". The TAM to be applied to e-learning as proposed in this paper thus includes self-efficacy, system accessibility, subjective norms, perceived ease of use, behaviour intention, and perceived usefulness based on the TAM as proposed by Park [16]. Such a model expresses the realization that these factors determine the attitude, behaviour, adoption and further intention of the user to technology, where further intention is affected by personal feedback of the user from external factors.

4 Experimental design

Using the TAM described, the author carried out a survey to evaluate the effect of cultural factors on technology acceptance in an e-learning environment. Two groups of students were surveyed (one each in Oman and the UK).

4.1 Student survey

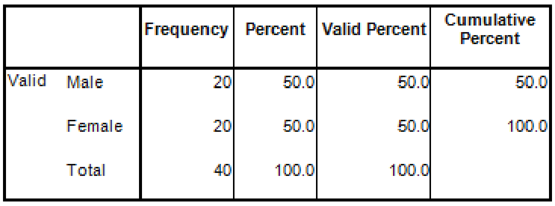
Target groups of students from Oman and UK were introduced to an e-learning model that took account of cultural factors. Suitable schools in the UK and Oman were selected on criteria that ensured general mutual comparability. Owing to various constraints at the time of the survey, a sample size of 40 participants (20 male and 20 female) was set for each country, as giving sufficient scope for data analysis and statistical analysis. The total of 80 participants presented a workable sample-size for capturing and analyzing data with a reasonable chance of representativeness. The two countries represent different cultures—western culture in a developed country (UK) and Middle Eastern/Gulf Arab culture in a rapidly developing country (Oman). It is evident from studies that educational practices—both teaching *and* learning processes—are tied to culture and tradition [31]. To members of the Middle Eastern and Gulf Arab communities, the challenge has been that the internet, ICT and therefore e-learning have arisen and are therefore inevitably associated with western contemporary culture, which is different from Islamic culture. In this regard it might be well to remember the dimension of 'uncertainty avoidance' owing to the Arab cultural feature of aversion to the unknown.

The questionnaire consisted of 20 questions and four student-profile questions. The questionnaire sought to elicit responses by offering a five-point Likert scale for each item: 1 (Strongly Disagree); 2 (Disagree); 3 (Neutral); 4 (Agree); and 5 (Strongly Agree) to measure the various e-learning technology acceptance variables. The questionnaire was divided into five sections. The first section had questions that provided a list of responses for the students to tick. These questions represented the demographic data (covering gender, age, cultural background, language, and experience with the internet). The second section measured social factors (the variables of language, qualifications, skills, and facilitating conditions). Section Three dealt with cultural factors that measured the variables of individualism/collectivism, uncertainty avoidance, and power distance. Section Four evaluated political factors, measured by the variables of use of social networks and social media. Finally, Section Five measured Technology Acceptance Model constructs, identified by the variables of perceived usefulness and perceived ease of use.

5 Results

5.1 Omani students—general

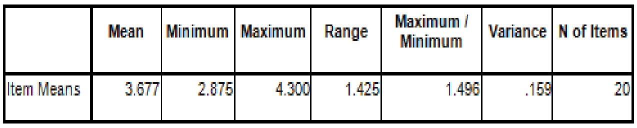
The questionnaire sample was selected from students in the tenth grade in different schools, and consisted of 20 males and 20 females (Table 1)



		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male	20	50.0	50.0	50.0
	Female	20	50.0	50.0	100.0
	Total	40	100.0	100.0	

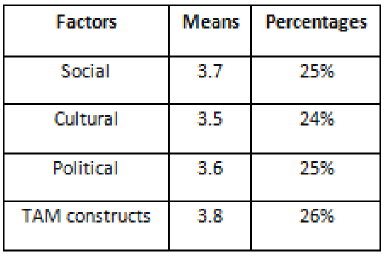
Table 1: Omani students' sample (in percentages)

The results showed that the average responses ranged between 4.3–2.8 of 5 (as shown in Table 2).



	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance	N of Items
Item Means	3.677	2.875	4.300	1.425	1.496	.159	20

Table 2: Phase 3 Omani students' sample: Summary Item Means



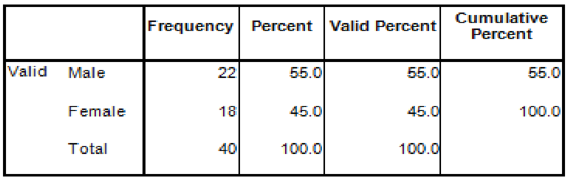
Factors	Means	Percentages
Social	3.7	25%
Cultural	3.5	24%
Political	3.6	25%
TAM constructs	3.8	26%

Table 3: Omani students' ranking of the four main factors

The students' responses reflected a high rate of acceptance of e-learning technology as seen in the items mean of 3.67, when all items contained in the questionnaire ranged between 1 (Strongly Disagree) and 5 (Strongly Agree).

5.2 UK students—general

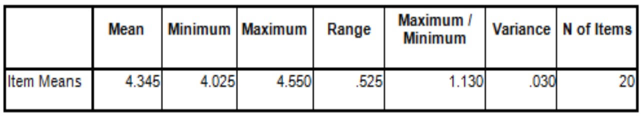
The research sample consists of students from Shorefields Technology College in Liverpool. The sample of 40 students comprised 22 males and 18 females registered for the academic year 2011/2012 as shown in Table 4.



		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male	22	55.0	55.0	55.0
	Female	18	45.0	45.0	100.0
	Total	40	100.0	100.0	

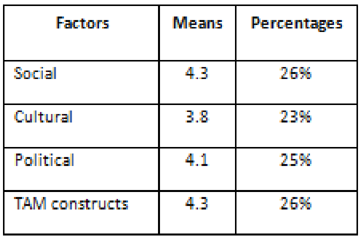
Table 4: Phase 3 UK students' sample (in percentages)

Questionnaires returns showed that the average responses ranged between 4.0–4.6 of 5 as shown in Table 5.



	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance	N of Items
Item Means	4.345	4.025	4.550	.525	1.130	.030	20

Table 5: Phase 3 UK students' sample: Summary Item Means



Factors	Means	Percentages
Social	4.3	26%
Cultural	3.8	23%
Political	4.1	25%
TAM constructs	4.3	26%

Table 6: UK students' ranking of the four main factors

The students' responses reflected a high rate of acceptance of e-learning technology as seen in the much higher items mean of 4.34, when all items contained in the questionnaire ranged between 1 (Strongly Disagree) and 5 (Strongly Agree).

5.3 Social factors

The comparison between the responses of Omani and UK students reveals a substantial discrepancy in only one aspect, that of language. Most of the relevant geography e-learning sites are presented in the English language. Others exist in a range of European and non-European languages but few are in Arabic. All Omani school-goers are taught English as a second language, but achieving consistently good results is a slow process [32]. As Omani students are normally exposed to learn English before any other language (such as Spanish, French, Chinese), the difficulties Omani students face are in the area of non-Arabic language content. This explains the particular discrepancy in the comparative lists of means of social factors (see Table 7). In all other aspects Omani students are seen to be close to their UK counterparts.

Means of social factors		
Factors	Cultures	
	Oman	UK
Language	3.1	4.2
Qualification/Skills	4.1	4.5
Facilitating Conditions	4.0	4.4

Table 7: Students' Responses: Means of social factors across two cultures (Oman & UK)

Omani students: Social factors came in *second* place in the ranking of the four main components of technology acceptance. As stated above, for Omani students the major problem resides in issues surrounding the language in which a particular e-learning website is presented. However, it is noteworthy that this problem has been highlighted by the *voluntary* responses of the Omani students themselves, when they could have allocated reasons for difficulties to other causes, to those outside their own competence or responsibility. It is encouraging to see that these students are prepared to be honest about their difficulties in using languages other than Arabic to search and surf e-learning websites—as is made clear in the responses to items numbers 1 and 11 (having means of 3.3 and 2.9 respectively).

In addition the qualification/skills are the important aspects of the e-learning environment where students need basic skills in using technology gained by studying the information technology (IT), as is clear from items 2 and 12. Facilitating conditions constitute another important factor as indicated in items 3 and 13, pointing to the importance of providing to schools greater numbers of computers and regular maintenance services to facilitate students' performance in the e-learning environment. Table 8 gives details.

Means of social factors		
Factors	Means	Mean of means
Language	3.1	3.7
Qualification/Skills	4.1	
Facilitating Conditions	4.0	

Table 8: Means of social factors according to Omani students' responses

UK students: Social factors came in *first* place in the ranking of the four main components of technology acceptance. As social factors comprise *language* skills and competences besides background qualification/skills and facilitating conditions, UK students possess an advantage that exists because of historical reasons. The vast majority of relevant websites are in English, and even though the teaching of modern languages has been in serious decline across the UK, students in the UK still have better opportunities for exposure to other languages (in the classroom and in real life) than do students in Oman. UK students have few or no problems when using and searching other-language e-learning websites, as is clear from response items numbers 1 and 11 (with means of 4.4 and 4.1).

Qualification/skills and facilitating conditions have high response-averages, indicating that the UK students agree with the importance of providing students with basic IT skills and/or IT study-courses to build their capacity in using and makes searches of e-learning websites. Furthermore, they believe that as long as the learning environment remains equipped at a suitable level for e-learning, this will help and support them in their learning. Details are shown in Table 9.

Means of social factors		
Factors	Means	Mean of means
Language	4.2	4.3
Qualification/Skills	4.5	
Facilitating Conditions	4.4	

Table 9: Means of social factors according to UK students' responses

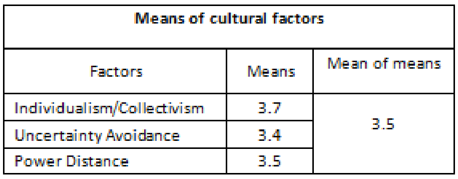
5.4 Cultural factors

Comparison between Omani and UK students in cultural factors reveals discrepancies in all aspects, whilst consistency in Omani responses is paralleled by a noticeable internal dichotomy within the UK responses. Uncertainty avoidance and power distance are stronger amongst Omani students. However, the individualizing effect of working on a computer (which introduces an isolating tendency effect amongst students) is offset in the case of Omani students by their stronger culture of cooperation; so that in many cases uncertainty avoidance and power distance can be mitigated, as Omani students will tend to seek help from their colleagues much sooner than their UK counterparts would consider doing so. Table 10 shows the comparative details.

Means of cultural factors		
Factors	Cultures	
	Oman	UK
Individualism/Collectivism	3.7	3.2
Uncertainty Avoidance	3.4	4.2
Power Distance	3.5	4.2

Table 10: Students' Responses: Means of cultural factors across two cultures (Oman & UK)

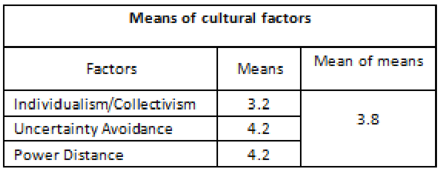
Omani students: Cultural factors (individualism *versus* collectivism, uncertainty avoidance and power distance) came in *fourth* place in the Omani students' ranking of the four main components of technology acceptance of technology acceptance. Omani students tend to cooperate and work as a team, in virtue of their strong background of living in a community in which the collective spirit of cooperation and interaction is still strongly alive, even in urban areas. This is shown by high level of agreement in response to the item "Using e-learning rather than traditional instruction creates an isolation atmosphere between me and my friends". They prefer to ask colleagues, friends and technicians when they encounter problems when browsing or learning from e-learning websites, as indicated in the means to response items 4 and 14. While Omani students experience anxiety in tackling unknown and unpopular learning websites—perhaps owing to a lacking of browsing skills or through aversion to the unknown—the mean of 3.4 is indicative that they still seek to challenge themselves in the e-learning context. As for power distance, in spite of the response mean of 3.4 for the proposition that using e-learning should be not limited to certain groups (such as mangers and highly qualified individuals), item number 6 clearly shows the trend of respondents to share experiences with their friends when they learn new topics from e-learning websites (a mean of 3.7). In fact these means indicate the problems that Omani students face in relation to this factor, showing their need for guidance and help to gain more confidence in using technology applications by providing more training and to counteract the notion that using technology should be limited to certain groups such as qualified individuals. Table 11 shows details.



Means of cultural factors		
Factors	Means	Mean of means
Individualism/Collectivism	3.7	3.5
Uncertainty Avoidance	3.4	
Power Distance	3.5	

Table 11: The means of cultural factors according to Omani students' responses

UK students: The same cultural factors came in *third* place in the UK students' ranking of the four main components of technology acceptance. Unlike Omani students, UK students show a high degree of reticence. The UK students did not think that the e-learning process creates isolation between them and their friends, as is clear in their low response mean in item number 4 (a mean of 3.1). However, their response regarding readiness to ask a colleague or friend in case of problems when browsing e-learning websites (item number 14) reflects an individualistic attitude regarding technical competence (and thus a reluctance to turn immediately for help in circumstances of difficulty). Thus when technology is available it tends to foment isolationist attitudes. Additionally, their replies illustrate a lower level of uncertainty avoidance—in response to statements "I prefer to surf the unknown and unpopular learning websites" and "I like to search for and explore new e-learning websites" (items means 4.1 and 4.4 respectively). Results also reflected weaker power distance effects in responses to "I prefer to share my experience with students and colleagues when learning new topics from e-learning websites" and "Using e-learning should not be limited to certain groups such as highly qualified individuals" (items means 4.3 and 4.1 respectively). However these results support Hofstedes' observations [3] regarding the lower ambient levels of uncertainty avoidance and power distance in western cultures. Table 12 shows details.

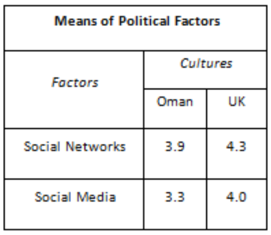


Means of cultural factors		
Factors	Means	Mean of means
Individualism/Collectivism	3.2	3.8
Uncertainty Avoidance	4.2	
Power Distance	4.2	

Table 12: The means of cultural factors according to UK students' responses

5.5 Political factors

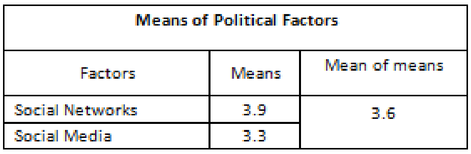
Comparison between Omani and UK students in political factors shows internal consistency in each set of responses. Young people in the west have been developing a sub-culture within each western society such that they have become differentiated at rapidly decreasing intervals—the so-called 'Generation X' (from roughly the middle to the end of the 20th century) and Generation Y (beginning by overlap near the end of the 20th century or beginning at the opening of the 21st century)—and these generations have become increasingly differentiated in their own characteristics. Thus, young people in the UK (as in other western societies) have had various means of proximate and distance interacting through social facilities that have been available for a long time. Consequently, a constantly developing cultural tradition about social interaction and networking has grown up amongst youth that is passed down the generations. Owing to historical background, the young generations in Oman have yet to develop a similar sub-culture within the larger society, and this is reflected in the ways in which they tend to use social networks and media, and in the patterns of use and purpose that characterize their habits.



Means of Political Factors		
Factors	*Cultures*	
	Oman	UK
Social Networks	3.9	4.3
Social Media	3.3	4.0

Table 13: Students' Responses: Means of political factors across two cultures (Oman & UK)

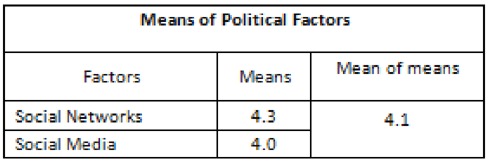
Omani students: Political factors (the use of social networks and social media) came in *third* place in the Omani students' ranking of the four main components of technology acceptance. For social networks, items numbers 7 and 17 had responses with means of 4.1 and 3.7 respectively, indicating that *social networks* such Facebook, Twitter, as well as various other forums and blog-sites using enhanced technology play a major role in providing and updating students with the latest news of political and other issues. In contrast, the mean of 3.3 for *social media* indicated the low incidence of using mobiles to follow political and other events. However political factors are now playing an increasingly important role in technology acceptance, but only to a certain extent because the government has yet to encourage greater freedom of discussion on various topics such as domestic political issues. This creates a certain amount of reluctance in the take-up of new technology, and it will take time before such a take-up can be stimulated further.



Means of Political Factors		
Factors	Means	Mean of means
Social Networks	3.9	3.6
Social Media	3.3	

Table 14: The means of political factors according to Omani students' responses

UK students: Political factors (the use of social networks and social media) came in *second* place in the UK students' ranking of the four main components of technology acceptance. Items numbers 7 and 17 had high means (4.4 and 4.3 respectively), indicating that the respondents believe in the important roles that *social networks* play in exchanging, providing and updating information on political and other issues. As for *social media*, items numbers 8 and 18 had the same response mean of 4.0 and this confirmed that UK respondents used internet via mobiles and YouTube to follow up political events and other issues. This usage reflects somewhat the tolerance of UK culture with regard to the discussion of political issues. Details are shown in Table 15.

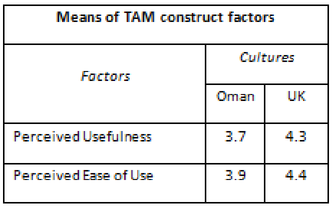


Means of Political Factors		
Factors	Means	Mean of means
Social Networks	4.3	4.1
Social Media	4.0	

Table 15: The means of political factors according to UK students' responses

5.6 TAM constructs

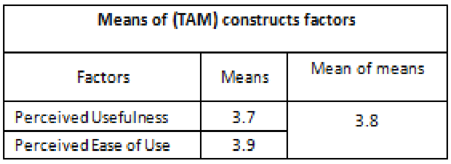
Perceived usefulness and perceived ease of usefulness were ranked in first place by both sets of respondents, although in the case of the Omani students, the majority in favour was not as high as in the case of the UK students.



Means of TAM construct factors		
Factors	*Cultures*	
	Oman	UK
Perceived Usefulness	3.7	4.3
Perceived Ease of Use	3.9	4.4

Table 16: Students' Responses: Means of TAM construct factors across two cultures (Oman & UK)

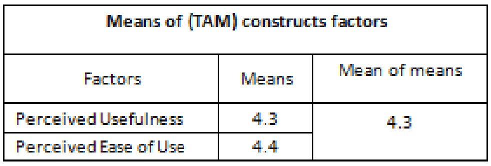
Omani students: *Technology Acceptance Model constructs* (perceived usefulness and perceived ease of use) came in *first* place in the Omani students' ranking of the four main components of technology acceptance. This response reflected acceptance of these two factors as providing to them many advantages, such as savings in time and effort, providing them with useful learning materials, and making it easy to find information from e-learning websites. However, the majority of the Omani student who accepted these constructs was not as great as that in the case of their UK counterparts. Details are given in Table 17.



Means of (TAM) constructs factors		
Factors	Means	Mean of means
Perceived Usefulness	3.7	3.8
Perceived Ease of Use	3.9	

Table 17: The means of TAM constructs factors according to Omani students' responses

UK students: *Technology Acceptance Model constructs* (perceived usefulness and perceived ease of use) also came in *first* place in the UK students' ranking of the four main components of technology acceptance. The UK students reported these two factors as being very important in using e-learning websites. Thus, they agreed that the use of e-learning websites helps to save time and effort, helps to find information easily and provides useful learning materials. Table 18 gives details.



Means of (TAM) constructs factors		
Factors	Means	Mean of means
Perceived Usefulness	4.3	4.3
Perceived Ease of Use	4.4	

Table 18: The means of TAM construct factors according to UK students' responses

6 Discussion

The detailed comparisons between the responses provided by the students from Oman and the UK indicate patterns that are generally what would be expected, given the historical background to the two countries and their educational systems. Within social factors, Omani students showed a particular disadvantage as regards ability to perform in English and other non-Arabic languages. English has become in many ways the *world language* for technology, and the Omani education system is acting to address this issue. Omani students are also seen as being at a disadvantage in two out of three cultural factors (uncertainty avoidance and power distance), but they are at a comparative advantage in the area of cooperation/collaboration owing to their cultural background. In political factors they are lagging behind UK students especially in their use of social media—the mobile media, especially—although as their society develops these differences are expected to decrease considerably. Given the responses regarding these three classes of factors (social, cultural, political) it should not be surprising that the majority of Omani students who accept the TAM construct factors is smaller than that among UK students. A lot of what is taken for granted by UK students is still new to students in Oman, although they are adapting rapidly to innovation. However, what might have been surprising would have been a majority of Omani students who did *not* view TAM construct factors favourably.

7 Conclusion

This study sought to develop an adaptive learning environment which is appropriate to the requirements of e-learning and the positive results that have been achieved are all reflected in the learning outcomes and the survey responses. The study was conducted through three experimental procedures, which obtained results that reflected positive attitudes among students towards using e-learning applications in terms of effectiveness and efficiency, and also gave different indications about the role of cultural factors in influencing the acceptance of e-learning.

The proposed TAM has four main factors to measure students' trend to use e-learning technologies and websites. Responses' mean of Omani students illustrated the place of cultural/social factors as TAM constructs, Social, Political and Cultural factors respectively. TAM constructs like usefulness and ease of use are represented the importance factors because the respondents used new experience and the interface and facilities of website are important to understand and used the technology components, therefore, this factor came at the first rank. Social factors came in second place in terms of the respondents' awareness that the facilitating conditions and qualification/skills are important factors to develop the effective learning environment where it seen to facilitating conditions as the recognition of the existence environment elements such as knowledge, time, financial resources, equipment, and access to hardware/software and these consider to be the initial requirements for any learning environment. In other side language is presented the problem with respondents as they found difficulties to search and surf in e-learning websites using other language. Political factors came in third place in terms of social media and social networks and their role to provide and update students for latest news of political issues. Cultural factors came in last place as students reflected a collective community and they have an anxiety from unknown and unpopular learning websites because they do not have enough skills to browse in internet or because they do not like to explore new e-learning websites. In addition, they have another problem regarding to their believe of using technology should be limited in certain groups such as qualified individuals and this illustrated their needs to be more confidence in using technology applications and to solve this problem they should provide them with more training to get necessary skills and abilities. UK students' responses mean in terms of the four main factors are ranked as TAM constructs/ Social, Cultural and Political factors respectively. UK and Omani students agree on the importance of TAM constructs factors which consist of ease of use and usefulness in terms of using e-learning websites. In contrast, unlike Omani students, UK students have not faced any problem with language when they search in English because it is their mother tongue. Unlike Omani students, UK students represented the individual community in using technology because of the availability of technologies networks, speed and services contrast to Oman culture. Results as well showed the less uncertainty avoidance and power distance within UK culture which support Hofsted's study [3] that approved the less effect of uncertainty avoidance and power distance in western cultures. Political factors presented the last place which means social media and social networks have played main role in exchanging, providing and updating the political issues according to UK respondents and this is reflected the tolerance of UK culture with respect to political issues following and discussing unlike Omani culture.

6. References

[1] Gay G (2000). Culturally Responsive Teaching: Theory, Research and Practice. New York: Teachers College Press. [see especially Chapter Two: 'Pedagogical Potential of Cultural Responsiveness'].

[2] Thomas F, Haddon L, Gilligan R, Heinzmann P, de Gournay C (2003). 'Cultural factors shaping the experience of ICTs: An exploratory review' [accessed June 2012 from http://www2.lse.ac.uk/media@lse/whosWho/AcademicStaff/LeslieHaddon/CulturalFactorsChapter.pdf].

[3] Hofstede G (1980). Culture's Consequences: International Differences in Work-Related Values. Beverly Hills, CA: Sage Publications.

[4] Del Galdo EM (1996). 'Culture and Design', pp 74–87 in Del Galdo EM, Nielsen J (eds.) International User Interfaces; New York: John Wiley & Sons.

[5] Hill CE, Loch KD, Straub DW, El-Sheshai K (1998). 'A qualitative assessment of Arab culture and information technology transfer'. Journal of Global Information Management 6(3):29–38.

[6] Collis B (1999). 'Designing for differences: Cultural issues in the design of www-based course-support sites'. British Journal of Educational Technology 30(3):201–216.

[7] Davis FD (1986). Technology Acceptance Model for empirically testing new end-user information systems: Theory and Results. Massachusetts Institute of Technology: Sloan School of Management: PhD thesis.

[8] Davis FD (1989). 'Perceived usefulness, perceived ease of use, and user acceptance of information technology'. Management Information Systems Quarterly 13(3):319–340.

[9] Davis FD, Bagozzi RP, Warshaw PR (1989). 'User acceptance of computer technology: A comparison of two theoretical models'. Management Science 35(8):982–1003.

[10] Hess TJ, Joshi K, McNab AL (2010). 'An alternative lens for understanding technology acceptance: An equity comparison perspective'. Journal of Organizational Computing and Electronic Commerce 20(2):123–154.

[11] Cornell RM, Eining MM, Hu P J-H (2011). 'The effects of process accountability on individuals' use of a familiar technology'. Journal of Information Systems 25(1):109–128.

[12] Fishbein M, Ajzen I (1975). Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research. Reading, MA: Addison-Wesley.

[13] Ajzen I, Fishbein M (1980). Understanding Attitudes and Predicting Social Behavior. Englewood Cliffs, NJ: Prentice-Hall.

[14] Venkatesh V, Morris MG, Davis GB, Davis FD (2003). 'User acceptance of information technology: Toward a unified view'. MIS Quarterly 27(3):425–478 [accessed May 2012 from http://csdl.ics.hawaii.edu/techreports/05-06/doc/Venkatesh2003.pdf].

[15] Venkatesh V, Bala H (2008). 'Technology Acceptance Model 3 and a research agenda on interventions'. Decision Sciences 39(2):273–315.

[16] Park SY (2009). 'An analysis of the Technology Acceptance Model in understanding university students' behavioural intention to use e-learning'. Educational Technology & Society 12(3):150–162.

[17] Arenas-Gaitán J, Ramírez-Correa PE, Rondán-Cataluña FJ (2011). 'Cross cultural analysis of the use and perception of web based learning systems'. Computers & Education 57:1762–1774.

[18] Suh C-K, Lee T-H (2007). 'User acceptance of e-learning for voluntary studies'. International Conference on Computer Engineering and Applications: Proceedings of CEA '07: Annual Conference 2007, pp 538–544.

[19] Lee Y-C (2008). 'The role of perceived resources in online learning adoption'. Computers & Education 50(4):1423–1438.

[20] Ziefle M, Jakobs E-M (2010). 'New challenges in human computer interaction: Strategic directions and interdisciplinary trends'. Full paper at the 4th International Conference on Competitive Manufacturing Technologies (pp 389–398). University of Stellenbosch, South Africa.

[21] Lim CP, Khine MS (2006). 'Managing teachers' barriers to ICT integration in Singapore schools'. Journal of Technology and Teacher Education 14(1):97–125..

[22] Teo T, Lee CB, Chai CS (2008). 'Understanding pre-service teachers' computer attitudes: Applying and extending the Technology Acceptance Model'. Journal of Computer Assisted Learning 24(2):128–143.

[23] Hofstede G (1990, 1997). Cultures and Organizations: Software of the Mind—Intercultural Cooperation and Its Importance for Survival. New York: McGraw-Hill [revised edition 1997].

[24] Srite M, Karahanna E (2006). 'The role of espoused national cultural values in technology acceptance'. MIS Quarterly 30(3):679–704.

[25] Ajzen I (1985). 'From intentions to actions: A theory of planned behavior', pp 11–39 in Kuhl, Beckman J (eds.) Action-Control: From Cognition to Behavior. Heidelberg: Springer.

[26] Ajzen I (1991). 'The theory of planned behavior'. Organizational Behavior and Human Decision Processes 50(2):179–211.

[27] Ajzen I (2011). 'The theory of planned behavior: Reactions and reflections'. Psychology and Health 26(9):1113–1127.

[28] Bandura A (1997). Self-efficacy: The exercise of control. New York: Freeman.

[29] Bandura A, Freeman WH, Lightsey R (1999). 'Self-efficacy: The exercise of control'. Journal of Cognitive Psychotherapy 13(2):158–166.

[30] Lau S-H, Woods PC (2009). 'Understanding learner acceptance of learning objects: The roles of learning object characteristics and individual differences'. British Journal of Educational Technology 40(6):1059–1075.

[31] Zhu C, Valcke M, Schellens T (2008). 'The relationship between epistemological beliefs, learning conceptions, and approaches to study: A cross-cultural structural model?' Asia Pacific Journal of Education 28(4):411–424.

[32] Sergon V (2011). 'Playing the Blame Game: English Education in Omani Government Schools'. ISP Collection Paper 1132 [http://digitalcollections.sit.edu/isp_collection/1132 accessed 20 May 2012].

ADEQUACY OF AN IMMERSIVE ENVIRONMENT AS A TEACHING-LEARNING TOOL APPLYING EXPERIMENTAL 3D LIMEN METHOD

Diego-Mauricio Torres-Arias and Trefftz, Helmuth.
NETSYS Institute and Universidad EAFIT

ABSTRACT

Immersive virtual environments or metaverse are representations of real or imaginative worlds created on the digital world with the purpose of making real situations or impossible experiences in the real world. Users on these worlds represent themselves via avatars that interact in representation of the user. At present, these virtual worlds are used for business processes, research and education. The purpose of this paper is to describe the process of using experimental 3D LIMEM method for adapting an immersive virtual environment to be used as a virtual teaching-learning tool of the English language as a second language.

Key words: immersive virtual environments, Avatar, metaverse, virtual reality, 3D LIMEM.

I. INTRODUCTION

In recent years, technological advances have been vital in the development of new paradigms of teaching in different educational areas. Approaching the classroom, teacher and knowledge to the student has been the focus of such developments. seeing how we move from the classroom to the web pages and hyperlinks, then the WebQuest, Wikis, forums, study groups, platforms such as Moodle and Blackboard, networks and knowledge in the cloud and others; these paradigms lead us to manage high levels of complexity, to live in diversity of opinion based on personal knowledge provided from a network as it narrates "the model for learning in the digital age" [2]. The next step in the evolutionary chain of virtual learning environments is the emergence of virtual immersive and semi-immersive environments within which we find a Unity (http://unity3d.com/), (http://www.vastpark.com/), OpenSimm (http://opensimulator.org/wiki/Main_Page), y Second life (Lindenlabs- http://secondlife.com/); being the last one the first in appearing as the sums of technology not only on 3D visualization and immersity, but also as integrated technology of chat, IP Voice and interaction in real time through customized avatars, with a high quality in the process of immersion to the student and giant possibilities in many fields, and the language study among them.

Having been launched in 2003 by Lindenlabs, SecondLife is the VIW (Virtual Immersive World) most active and used in higher education, in which universities and other education centers make active presence [3]. An example of this is offered by recent research focused on sociological, psychological and economic aspects that are focused taking advantage of SecondLife as a support to people with social challenges such as overcoming social anxieties and fears, or testing the grade of satisfaction of participating on SecondLife according to their gender [5]. In the case that concerns us, languages, these already exist some projects, organizations, and schools on SL (SecondLife) dedicated to the teaching of English applying traditional philosophies such as constructivist and even creating new paradigms as dogma or SLOODLE, among them we find AVALON - http://www.avalonlearning.eu - and NIFLAR -http://www.niflar.eu – (experimental worlds for the language teaching of English in 3D), KAMIMO PROJECT (Island on SecondLife) ASIMIL8, VIRTLANTIS (island on SecondLife to the study of a second language), TEACH YOU TEACH ME (island on buddy network - http://tuwien.esnaustria.org/ -) and EDUNATION (island on SecondLife to the learning of languages). In the area of educational projects for example, AvatarEnglish.com as school of learning. Edunation II and III, special island where there are located different teaching centers; and educational institutions paths of great global relevance as the Instituto Tecnológico de Las Américas (ITLA), Harvard, Massachussets Institute of Technology (MIT) and Stanford University, and in the colombian case, Los Andes University. These facts indicate us that the implementation of these new technologies in the field of teaching-learning “isn’t a technological problem at all, as a social challenge that require an educational solution” [8].

Thus, some visionaries seek to incorporate the new ICTs and the 3D world to generate a new educational revolution with greater force in the field of the second language teaching, a fact thateach day is taking importance for the future of students and professionals. However, none of the previous efforts, despite of the innovative nature of its works, have focused on making a measurement of the quality of the education and success of the strategies implemented to effectively make the student learn a second language more comfortably, in less time and in an easier and more affordable way.

The purpose, then, of the project described in this paper, is to perform an evaluation of the level of absorption of information learned through an ICT environment, remote and immersive 3D such as SecondLife (SL), in juxtaposition with classical classroom teaching in the classroom area using experimental 3D LIMEM method [14]. This is because there is neither relevant nor solid information about it and on which can make decisions effectively and justify huge expenses on virtual educational centers, islands (in this context, an island is a virtual portion of the land which is sold by SecondLife to an entity or real client, which builds an environment using 3D development tools) or huge invests of universities on SL. In this paper is described the process of adequacy of the environment used to these jobs according to LIMEM 3D METHOD guidelines.

II. RELATED WORKS

Among the many related works to the use of immersive 3D learning environments, several are highlighted internationally, efforts of different universities and centers of study that explore new technologies for teaching-learning, this is the case of Harvard University, the New York one or MIT one, all of them highly internationally, which offer to their students educational interactions on 3D virtual environments. Within these institutions stands with Harvard University with its course on Secondlife called: "CyberOne: law in the court of public opinion", which is designed specifically for law school and its Extension School. There exist other courses on virtual environments offered as academic experiences such as: HRSA Play2Train, which simulates a hospital environment in which students must attend different types of emergencies using their avatar. In Colombia, universities such as EAFIT make research works on the SecondLife platform through an island called Telling East (EAFIT Colombia - Land 6144 sq / Corner Ocean Front / Class 5/Mature/Terraform, Tellings Beach East), a place where students have the opportunity to try constructions on 3D or in the case of SENA whose main island is SENA Colombia 3d (aaSENA 3D virtual SENA), and which has built a great center to show to the world the Colombian coffee, and a virtual hospital and a virtual farm. College Minute of God has also taken significant research on the subject, highlighting its work: "3D Virtual Environments, pedagogical alternative to the Foster of the Collaborative Learning and Management Knowledge in Uniminuto "investigation conducted by Juan C. Franci Quinche and L. Gonzalez, and which is intended to lay the groundwork for a future virtual Uniminuto using Sloodle experiences [6].

We also find universities like the Carlos III University, Seville University, Rosario University, Aveiro University, Navarra Public University and do Porto University among others.

So far, in the directory of Second Life's owners there are more than 95 educational institutions.

However, all these efforts are aimed at building a site to measure or develop a specific program unlike our research which neither aims to build an educational world through Second Life metaverse, nor create a complete academic program to be followed, instead, what it is wanted to get is an evaluation of the effectiveness of using 3D immersive virtual worlds when executing a teaching-learning interaction of a second language compared to the traditional learning environment in the classroom.

To achieve the goal we propose a process outlined by the following steps, based on the experimental method 3D LIMEM, developing and design guidelines and implementation of e-activities in 3D virtual worlds [12] raised in the VI conversatorio 2011 in Secondlife:

III. CONTEXT

3.1. Definition the subject of study

For effects of recreating the best way the development of a teaching-learning interaction must choose a specific thematic but also with an intermediate degree of complexity, which can be developed in a reasonable time, which is commensurate with the student level (A2 in the Common European Framework scale) chosen for the process and that is part of the academic program of an institution with a certified program. Taking into account these characteristics, the thematic of present simple was chosen, specifically the use of the third person rules (use of inflection in verb endings and related auxiliaries), through the context of describing the daily routine of a student and building or answering questions about it. Therefore, it was necessary to design a method which has been called LIMEM 3D (Learning immersive 3D environment Method), build a blog based on the thematic to be taught and to adapt a simple environment of interaction with the student through Second Life, using a portion of the 3D land of the Universidad EAFIT.

3.2. Selection of the virtual platform

One of the features of 3D immersive environments is that the users are represented through avatars. The user "lives" that world through a virtual "I" that is called avatar "[1]. Additional to this, an immersive virtual environment must guarantee us the characteristics of Corporeity of the avatars and the environment as minimum. Interactivity between avatars and with the elements of the virtual world and finally persistence; in other words, not to disappear when going out the environment. Taking into account these characteristics and preliminary design of the experiment were evaluated 4 tools, they are: Open Sim, Second Life, Unity and Vastpark which compare their characteristics and is chosen the one with the best score obtained according to the case of use raised for the test in question. For that there has been used the Virtual tool World Finder, from the page: http://www.daden.co.uk, which offers the possibility to compare the 4 platforms listed according to the specific selection of parameters.

The consultation on that page showed as result that the appropriate environment was SecondLife, here there is attached the list of chosen parameters and the score obtained for each platform:

Can build and customise in-world:	***(rather than use an external 3D design tool or ask us to change everything)***
Highest possible quality graphics:	*(yes this is a use case not technical requirement, and note that not virtual world will currently match the best computer game)*
NEED TO MODEL A LARGE AREA (> 1 sq km):	*N/A*
Custom client:	*(e.g. no vendor branding, only functions needed by user etc)*
Engagement mode:	*(This is about users per activity - it is assumed that there will always be multiple concurrent users)*
Runs in browser (with plugin):	*(no 3D virtual world runs yet in a browser without a plugin)*
Runs on mobile:	*N/A*
Deploy on own servers:	*(i.e. behind firewall to keep data secure)*
Robustness/security of code:	*(i.e. minimal chance of code failures or exploitation)*
Import of existing 3D models:	*(e.g. from Collada, 3D Studio max etc)*
Lowest Cost Possible:	*N/A*
Security of Supply:	*(combination of size of user base and likelihood of vendor still being in business in 5 years)*

N/E: Not Apply.

The best match for your requirements is Second Life with a score of 257

Second Life scored 257
Unity scored 243
OpenSim scored 245
Vastpark scored 217

As additional argument, the MVI more active and used in higher education is SecondLife (SL), launched in 2003 by Linden Lab2 where universities make presence not only as a pedagogical tool, but also to make known to the world and attend to different educational plans [3]. Furthermore, this platform was chosen because its physics engine is advanced (Havok 4), users can hear and see the video and audio streaming within SecondLife which is compatible with MPEG audio and Ogg Vorbis formats (it is required to install Quicktime).

The negative point is the necessity of the machine, which to run smoothly with the platform must fit with the minimum characteristics required by www.secondlife.com. And they are:

Internet Connection: Cable or DSL Cable or DSL Operating System: XP, Vista or Windows 7. CPU with SSE2 support, including Intel Pentium 4, Pentium M, Core or Atom, AMD Athlon 64 or higher. 1.5 GHz (XP), 2-GHz (Vista) 32-bit (x86) or higher. RAM: 1 GB or more. Screen Resolution: 1024x768 pixels or higher graphics card and XP or higher operating system.

IV. THE EXPERIMENTAL 3D LIMEM METHOD.

Experimental 3D LIMEN Method, is a useful tool for teaching-learning interactions through 3D environments. It is actually developing and has four possibilities in its application, as is showing:

A. Learning path: This branch of application intends built a path through learners understand the suggested topic. This method application uses four different windows in order to present the thematic to develop (with their principal aspects), presentation of topic into a context or through character using the principal aspects, present a route with activities to develop during it within the 3D world - chosen environments) and present Links test about thematic chosen.
B. Simulation: this branch focuses on simulate real situations, with complete details, into a controlled context through avatars and developed 3D sceneries in order to analyze situations in a deep way. This branch is focused on reactions and final results.
C. Emulation: Through this line, 3D LIMEM Method applies design and function of machines (Simulate physical machine aspects and emulate functions, shape, states and interactions). It is useful for training in machines uses or usability test.
D. Role play: it is similar to Simulation; however, the difference is that every role is specified by researchers for every single participant in order to find a particular result that is previously proposed as hypothesis. It branch is focuses on interactions and final results.

4.1 Adequacy or searching of appropriate environments for the development of the thematic to be implemented

Depending on how the environment is adequate and be developed, this will be attractive or not to the student (in this case user), which "receive plants of implementation in small groups. Instructors use these plans to determine the work on Secondlife and the selection of learning tools that students, as a group, are introduced according to their position" [10].

SecondLife is based on a metaphor of real estate located in an ocean, in which the users can buy islands. These islands can be "developed", which means that any 3D artifact such as buildings, cars, clothes, etc ... can be built using scripting language for SecondLife [5]. However, the use of environments already recreated and that fit to the purpose of teaching and a simple construction, and that have similar characteristics to those of a virtual classroom such as: using interaction tools such as discussion forums, live chats and email, and that offer us certain advantages as: a more friendly environment for users, ease of configuration and the use of integrated tools (email, forums, chat, etc.). Integration on Internet. Management of a single program: the browser. Ease of adapting of content [11], are sufficient to obtain good results. It is so to set a section within an island we use prims (primitive object) or primitive (generally cubes extracted from an environment). Keep in mind that each island has an area of 512 m2, and that has a total of 15,000 prims (on SecondLife the term "prim" is used to refer to primitive objects. Except your avatar (AV or AVI), the floor of the simulator and the trees, all the objects on SecondLife are made of Prims. - http://wiki.secondlife.com/wiki/Acerca_de_los_Prims) to build approximately 117 new objects.

4.2 Defining shape, size and content of the screen Prims for our experiment.

According to experimental 3D LIMEM METHOD, for this experiment we need to build four screens previously designed each one for a specific function and all of them represented on a blog. To start the process of its construction we can use primitives from the 3D environment and use the incorporated tools in order to model the required screens.

We can also make constructions in applications such as SIMM ON A STICK, which simulates a 3D environment and permit to convert my team on served and user, build and export these constructions. Use OpenSimm, open Crocket or other creating programs of 3D objects that allow us to export or do rezzing (create or put something from your inventory to the virtual world) such as Maya or 3D MAX, as well as use public Sandboxes (places where you can build and then make a copy of the work to save in an inventory or apply on an avatar or island) to make buildings to be reused. However, when choosing this option it must take into account that Secondlife charges us for calculating the amount of object prims we want to load.

Additionally, there is a large market of components (reusable buildings or objects) that are readily available via the web or on SecondLife Marketplace (https://marketplace.secondlife.com/) and other related pages.

4.3 Using the created screens according to experimental 3D LIMEM Method.

For this adaptation of specific environment we rely on the concept of active learning route proposed by us in the experimental method in development called 3D LIMEM, a strategy which in this case proposes the implementation of 4 screens based on the development of a blog (http:/ / sleedch.blogspot.com) or web page about the thematic chosen to treat. Each one of the 4 windows must fulfill with a specific mission within the teaching-learning context planned as: window 1 (presentation of the language thematic to develop with its corresponding grammatical aspects), window 2 (presentation of a context or character with base on which to develop the thematic), window 3 (Trip of the learning route and activities to develop during it within the 3D world - chosen environments) and window 4 (Links with the thematic test). Here there are attached the windows models built and the blog base (Figure 1):

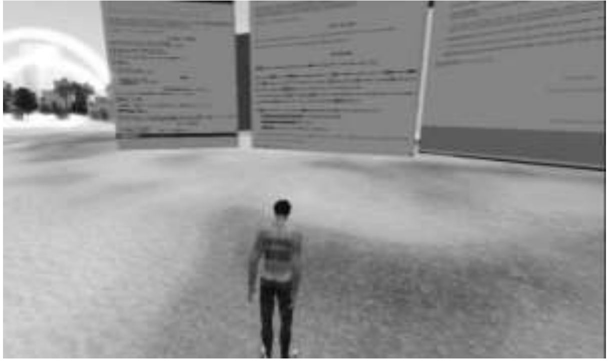
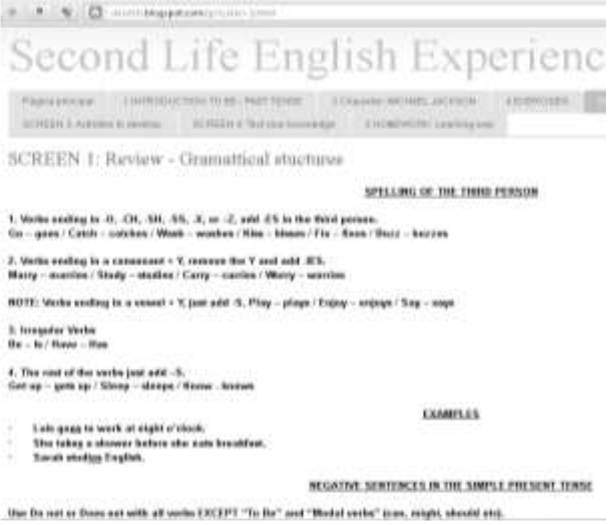


Figure 1: Created Screens at EAFIT – SecondLife

The final result of the LIMEM 3D method implementation for this experiment generates the following implementation for each window:



Second Life English Experienc

SCREEN 1: Review - Grammatical structures

SPELLING OF THE THIRD PERSON

1. Verbs ending in -O, -CH, -SH, -SS, -X, or -Z, add -ES in the third person.
Go – goes / Catch – catches / Wash – washes / Kiss – kisses / Fix – fixes / Buzz – buzzes

2. Verbs ending in a consonant + Y, remove the Y and add -IES.
Marry – marries / Study – studies / Carry – carries / Worry – worries

NOTE: Verbs ending in a vowel + Y, just add -S. Play – plays / Enjoy – enjoys / Say – says

3. Irregular Verbs
Be – Is / Have – Has

4. The rest of the verbs just add -S.
Get up – gets up / Sleep – sleeps / Know – knows

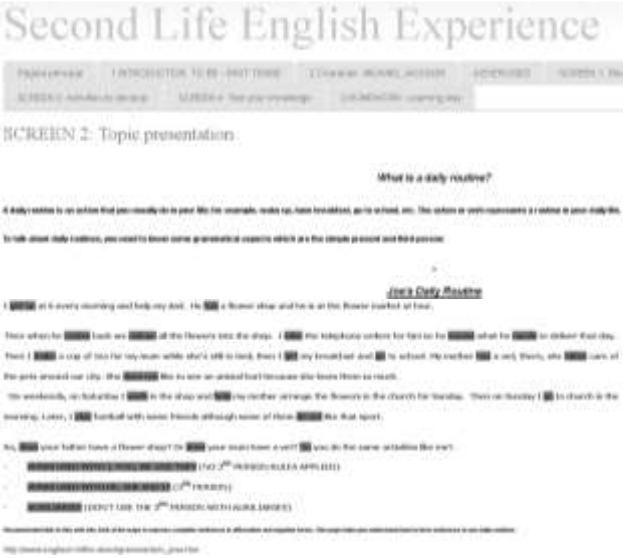
EXAMPLES

- Luis goes to work at eight o'clock.
- She takes a shower before she eats breakfast.
- Sarah studies English.

NEGATIVE SENTENCES IN THE SIMPLE PRESENT TENSE

Use Do not or Does not with all verbs EXCEPT "To Be" and "Modal verbs" (can, might, should etc).

Figure 2: Window 1. Third Person Rules. Grammatical Structures Review (http://sleedch.blogspot.com/p/screen-1.html)



Second Life English Experience

SCREEN 2: Topic presentation

What is a daily routine?

[illegible]

Joe's Daily Routine

[illegible]

Figure 3: Window 2. Topic Presentation, Daily Routine. (http://sleedch.blogspot.com/p/screen-2-topic-presentation.html)

Second Life English Experience

SCREEN 3: Activities to develop

[illegible]

Figure 4: Window 3. Activities to be developed, places visit and expected products.
(http://sleedch.blogspot.com/p/screen-3-activities-to-develop.html)

Second Life English Experience

SCREEN 4: Test your knowledge

Develop every single exercise in order to practice all the concepts learned: Third person, simple present and daily routines.

Third person exercises: [illegible]

Present simple exercises: [illegible]

Short answer: [illegible]

Negative form: [illegible]

Daily routine: [illegible]

Listening daily routine exercises: [illegible]

Figure 5: Window 4 Testing their knowledge, exercises (http://sleedch.blogspot.com/p/screen-4-test-your-knowledge.html)

NOTICE: you can see the original contents on the blog through: http://sleedch.blogspot.com/

In the window 1 is intended that the student has the summary of grammatical elements to be learned and to be used from the second language.

The window 2 introduces the grammatical elements seen in window 1 through application of them within a context in particular.

The window 3 proposes to visit different places within the immersive world in order to carry out various activities that will be tested and used as evidence in the training process.

Finally, the window 4 proposes a series of pages through which we can, through a test, evaluate the new knowledge learned.

The learner interaction with the screens is done throughout the whole process, either as a consultation (Screen 1), as applied example (screen 2), and calendar of activities to follow (Screen 3) or as part of the training of the learning (Screen 4). During the tour given on screen 3 students will generate deliverables that his/her teacher checks at the same time and to be able to keep constant communication via chat, teleport to a particular place in real time or use of email.

V. CONCLUSIONS.

For its technical characteristics of interaction capacity with the user and with other 3D platforms, the robustness of its development which includes excellent levels of security and continuous updates of its visor, as well as its imitation engine Havok 4, SecondLife becomes the tool for excellence for educational labor on 3D environments.

According to Experimental 3D LIMEM METHOD for performing a full teaching learning process does not necessarily require a complex development within a virtual world, adapting the environment to its purposes is useful. Take what already exists and what is functional to the educational process proposed, as long as there exists a previous planning process and clear learning objectives, a proper training in the management of the platform to be used and the technological means for its ideal operation. Make only that you can find.

Making changes to the surrounding environment, import objects or even build on 3D immersive environments is not a technically demanding work given the advances in panel tools available for these effects.

The success of the technical implementation of teaching-learning process of a 3D immersive environment is based more on its previous planning than in the manipulation of the available items, and in good internet connection and hardware component that guarantees the fluency of the expected interactions.

The field is promising for the future and requires more research processes that reinforce the proposed hypothesis in this article. In a first article based on the same research it is presented the final results of the application process [13].

REFERENCES

[1] Baños G., Miguel, Rodriguez G., Teresa C. 2012. V Simposio Las Sociedades ante el Reto Digital. Universidad del Norte. Bogotá, Colombia.

[2] Siemens, George. 2007. Conectivismo: Una teoría de aprendizaje para la era digital. Traducción: Diego E. Leal Fonseca.

[3] Baker, S. C., Wentz, R. K., and Woods, M. M. 2009. Using Virtual Worlds in Education: Second Life® as an Educational Tool. Teaching of Psychology.Volumen 36, No. 1. ISSN-0098-6283.

[4] RAMOS NAVA, María del Carmen, Larios Delgado, José, Cervantes Cabrera,Daniel y Leriche Vázquez, Renato. “Creación de ambientes virtuales inmersos con software libre”. Revista Digital Universitaria [en línea]. 10 de junio 2007, Vol. 8, No. 6. [Consultada: 11 de junio de 2007]. Disponible en Internet: ISSN: 1607-6079.

[5] Strobel, Johannes, Hawkins, Conrad. Designing in Second Life: Identity Construction and Learning in a Virtual Informal Environment. Journal of Online Engineering Education, Vol. 1, No. 1, Article 2. School of Engineering Education, Purdue University, USA.R. W. Lucky, “Automatic equalization for digital

communication," Bell Syst. Tech. J., vol. 44, no. 4, pp. 547–588, Apr. 1965.
[6] Quinche, Juan C., L. González, Franci. 2011. Entornos Virtuales 3D, Alternativa Pedagógica para el Fomento del Aprendizaje Colaborativo y Gestión del Conocimiento en Uniminuto. Corporación Universitaria Minuto de Dios, Bogotá-Colombia.
[7] García R., Miguel A., El-Seoud, Samir A. y otros. 2008. Collaborating and learning a second language in a Wireless Virtual Reality Environment. Int. J. Mobile Learning and Organisation, Vol. 2, No. 4.
[8] Garrison, D.R., Anderson,T. 2005. El e-learning en el siglo XXI: Investigación y práctica. Octaedro Barcelona España.
[9] Castañeda, Patricia, Alvarez, Gloria P. y otros. 2011. Informe Proyecto Pertinencia y condiciones facilitadoras del uso de Ambientes Inmersivos Virtuales para mediar procesos de aprendizaje colaborativo, en un caso de la asignatura de Ética y Colombia Realidad e Instituciones Políticas CRIP. Escuela Colombiana de Ingeniería. Universidad del Rosario. Colombia.
[10] Liski, Riitta, Syren, Irma. 2011. Experiences of operating and studying in second life: conclusions for training design. Turkish Online Journal of Distance Education-TOJDE August 2011, ISSN 1302-6488. Volume: 12 Number: 3/2 Special Issue on Second Life Applications in Distance Education Article 3.
[11] Mendoza, Hellyss. Modelo de evaluación de plataformas tecnológicas virtuales. Escuela de computación Valera - estado – Trujillo. Universidad "Valle del Momboy". Venezuela.
[12] Rodriguez, Teresa, Baños, Miguel. 2011. Secondlife: una herramienta para la mejora de las dinámicas de comunicación y aprendizaje en la enseñanza online. VI conversatorio de 2011 en Secondlife. Grupo Ciberimaginario. Universidades San Martin de Porres y Rey Juan Carlos. España.
[13] Torres-Arias, Diego_Mauricio, Trefftz, Helmuth. 2013. Educational effectiveness of using a shared virtual immersive environment for teaching english as a second language. Universidad EAFIT, Colombia.
[14] Torres-Arias, Diego_Mauricio. 2014. Método 3D LIMEM. Instituto Netsys, Colombia.

Author:
Diego Mauricio Torres Arias.
Data Systematization Technologist, Antonio Nariño University, 1995.
Systems Engineer, Antonio Nariño University, 1998.
Technological Specialization in Project Management, SENA, 2011.
Degree in Modern Languages, University of Quindío, 2012.
Magister in systems engineering and informatics, EAFIT University, 2013.
Specialist in management for engineers, Pontificia Bolivariana University (UPB), 2014.

Co-author:
Helmuth Trefftz, Ph.D.
Head, Department of Computer and Systems
Chief, Laboratory of Virtual Reality
EAFIT University.

SESSION

WEB BASED INSTRUCTION, ONLINE SYSTEMS AND METHODS, DISTANCE LEARNING, MOBILE SYSTEMS

Chair(s)

TBA

Research Directions for Teaching Programming Online

Amber Settle[1], Arto Vihavainen[2], and Craig S. Miller[1]
[1]School of Computing, DePaul University, Chicago, Illinois, USA
[2]Department of Computer Science, University of Helsinki, Helsinki, Finland

Abstract— *Online education has a long-standing tradition in academia, and yet online programming remains a relatively undeveloped area in the computing education literature. This is in sharp contrast with in-person programming courses, which have been a favorite subject of study in computing education. Research into teaching on-site programming is so extensive that numerous practices have emerged and are typically supported by instructional theory, empirical results or both. In this article we identify some of the commonly accepted practices for teaching programming in an on-site environment and survey the work that has been done for online programming. In doing so, we contrast the rigor of on-site programming research with the relative immaturity of educational practices for online programming. We identify research questions and future directions for online programming educators, with the goal of inspiring the same high-quality work that on-site programming research has produced.*

Keywords: online programming, programming, pedagogy, tools, best practices

1. Introduction

The computing community has recently witnessed a rebirth of interest in online education. The immense popularity of Khan Academy and Code.org, and the collaboration between organizations like Coursera, edX, and Udacity and prestigious universities like MIT, Harvard, Stanford, and others have brought new energy to a long-standing educational approach. The attention from the academic community, for example in the creation of new conferences like ACM Learning@Scale, and the interest from the popular press have converged to produce diverse opportunities for reaching hundreds of thousands of students.

While the new possibilities have energized many educators in computing, lost in the enthusiasm is the acknowledgment that online education, often called distance education prior to 2000, has existed for decades. To provide just two examples of enduring online education journals, *Distance Education* was founded in 1980 and *The American Journal of Distance Education* was created in 1987. What the recent initiatives arguably contribute is large scale access to students, a new and highly interesting development in online education. There is also a much stronger focus on teaching programming in the latest round of online educational initiatives.

Despite an established body of research in online education, research in teaching programming online remains underdeveloped. Unlike traditional programming courses where scores of researchers have conducted numerous quantitative and qualitative studies of how best to teach students to program (for surveys, see [55], [49]), the research into teaching online programming rarely identifies replicable practices that are either theoretically motivated or supported by empirical study. The majority of published papers are experience reports, and such articles are difficult to adapt and generalize. This leaves educators working on teaching online programming today with little guidance in how to approach their work. However, unlike previous online education research, the stakes now are higher. Many schools are adopting programming courses into their offerings, often basing their content on existing online courses. The attention of the press may lead any mistakes to be widely reported, and momentum in reaching hundreds of thousands of interested students will not be easy to recapture.

In this article, we explore the space of traditional "on-site" or "in-person" strategies for teaching programming to understand how it is that educators have approached the problem, after which we outline research done specifically on online programming. By "online", we mean a course in which a large portion of the activities and assessments are completed by students who are not on-site. In doing so we identify a number of gaps between the work in teaching on-site and online programming and provide suggestions for how to advance research in online programming in a more methodical and rigorous direction. Doing so will allow online educators to make more significant progress than previous researchers in order to capitalize on its momentum.

2. Teaching Programming

Teaching and learning programming has been researched widely during the past few decades to the point where standard educational approaches are forming from the numerous experiments. In this section, current educational practices that are used in teaching programming are discussed, with the goal of emphasizing the increasing maturity of the area. While a comprehensive survey is not possible here, a range of practices is presented, selected for common acceptance and diverse representation. Although the practices are presented separately, many of them have been studied and applied together to form hybrid courses (see e.g. [50]).

2.1 Changing Interaction

Classrooms have been traditionally seen as large auditoriums filled with thousands of students. The most stereotypical understanding of this situation is the classroom with a lonely teacher talking to students who struggle to follow. In response, computing education researchers have identified numerous ways in which students can be more fully engaged in learning to program, either by increasing student to student interaction, student to instructor interaction, or student to content interaction [45], [1]. Many of the approaches use active learning, where students not only do, but think about what they are doing [7].

Peer instruction [42] was originally created to solve the passive classroom-issue in physics education [19]. In order to have better participation from students, they prepare for lectures beforehand by reading provided material. As students have already read the material, the lectures are devoted to the teacher discussing the content, and engaging the students with interactive questions from which both the teacher and students benefit. Although some studies suggest that quizzes in lectures are not efficient [43], the questions in peer instruction are done in multiple iterations where the students think both alone and collaboratively in groups. Peer instruction helps instructors adapt their class to address student misconceptions, increase student retention, and improve learning outcomes over traditional lectures [19], [56].

Another approach to tackling the passive classroom is to move to hands-on programming assignments as soon as possible [67]. Students benefit immensely if instruction is moved from large classes to smaller classrooms with integrated labs and hands-on activities [69], [10], [67]. Methodologies such as lab-centric instruction [15], [62], [63], [39] and studio-based instruction [13] provide ways in which this can be done; typically students work on tasks under guidance from instructors and peers that provide support for the students if needed.

2.2 Collaboration and Peer Support

In order to have multiple directions for input, peer reviews can be utilized given that the guidelines emphasize the learning objectives [4]. Peer evaluation is often used in studio-based classroom settings [31] and, in one context, the quality of students' reviews have been found comparable to those provided by paid teaching assistants in computer science [29].

Work in laboratories can be done in pairs or in small groups [69] and organized to emphasize good programming practices such as the ones outlined by extreme programming [57]. The most typical extreme programming practice taken into introductory programming laboratories is pair programming, where pairs of students work together to create programs. As two students work on one assignment, the work is constantly being reviewed, and possible challenges can be discussed directly with the peer. In various studies, pair programming has been found to improve students' performance and retention [47], [11].

Additional collaboration and peer support can be facilitated by recruiting students as teaching assistants and mentors. In peer-led team learning, students work in groups that are led by e.g. peers or undergraduate teaching assistants, who are typically trained and supervised by faculty [30], [32] and are also themselves learning during the process. Peer-led team learning reworks student interactions significantly, although it is evident that recruiting the correct people and providing them proper guidance is of the utmost importance. Peer-led team learning has been reported to increase student retention [58] as well as increase the amount of students that choose CS as a major [46]. In addition, the use of undergraduate teaching assistants has been reported to create a more enjoyable context [20] as well as to help students improve their teaching and mentoring skills [68].

2.3 Modeling Problem Solving

When students are learning to program, they benefit immensely if they have a model on how a programming problem should be solved. Some modeling can done by process recordings [6] and by task design (see e.g. [65]), while for some, the use of a virtual classroom can be beneficial [8]. One suggestion is to utilize case studies [40], which consist of a problem and a narrative description on how solutions for the problems have been reached; case studies can be also seen as worked examples [14].

Lectures can be also used for worked examples [14], during which the teacher discusses a problem and works through it in a step-by-step fashion. Extensive empirical study has shown that novices greatly benefit from seeing worked examples in the early stages of learning [37], [44]. Worked examples are one approach for providing students a mental model, an explanation of the thought process that is needed to solve a problem. The need for a mental model is emphasized in teaching approaches that rely on cognitive apprenticeship, which is a theory on the process of how a more experienced person can teach a skill to a novice [16]. Cognitive apprenticeship has been adapted and discussed in the programming instruction-context (see e.g. [2], [14], [67]).

2.4 Engaging Content and Contextualization

Activity in classrooms can also be increased via the use of engaging content. As an example, in media computation [27], [24], students are introduced to computing using an interesting context; multimedia. The lectures often evolve around live-coding [3] examples, where the lecturer creates representations such as music or graphics. Media computing has been reported to increase student retention across various tertiary education institutions [28].

The way the material and assignments are designed play a large role since they form the main media with which students spend their effective learning time. As no material fits all contexts, it is important that it is contextualized so that the students can relate to it (see e.g. [64]) and that it supports the incremental steps in learning to program.

2.5 Adaptive Teaching

Adjusting the difficulty level of a course to match the students' ability level is important. Many authors suggest taking an incremental approach to building the material, which allows students to slowly build up their knowledge and skills (see e.g. [67]). This can be reflected both in the tasks that the students do, where assignments can be split into smaller parts that form subgoals, and also in the material, where one can e.g. emulate mini-languages by first only focusing on a small subset of a language [12].

Mini-languages typically have a small syntax and simple semantics, which makes starting programming easier. Mini-languages have been combined with visual programming environments [17]; the use of environments such as Scratch for teaching at-risk students has been noted to increase success in introductory programming courses [53].

3. Teaching Programming Online

As we have already noted, distance-education has a long history of study. At the same time, as our review illustrated, the study of teaching programming on-site has produced a variety of practices that are well supported both theoretically and empirically. In contrast, the study of online programming courses is still underdeveloped in the literature. As an example, performing a query[1] for articles on the ACM Digital Library related to teaching programming online using keywords "(programming OR cs1) AND (online OR distance) AND (teaching OR education)" returns 159 results, while the query "(programming OR cs1) AND (teaching OR education)" returns 1850 results.

Only a small part of the 159 articles contain somewhat relevant results, while still outlining mostly experience reports. Only a handful sought to identify underlying causes. Examples of the experience reports contain a description of how a consensus on online course structure was formed [52], what observations educators received from using specific platforms [51] and languages [60], and how the capabilities of web (i.e. no need for downloading software, dynamic sites) can be leveraged to bring more value to students [72] and to increase collaboration [34]. Some discussion on how to design and implement materials to support students learning to program exists as well, whether in an online environment or as a supplement to a more traditional course [22], [33].

Various studies have considered differences between students in local and online classrooms. While learning outcomes are often highly related to student motivation and reasons for participation as well as course outcomes, a meta-analysis on online learning that also included non-programming-related articles suggests that instructor-directed and collaborative instruction provides better results than independent study, and that the practice of providing quizzes does not seem to be more effective than assigning homework [43]. One should also remember that online participants are often adult students who cannot attend the traditional classroom environment due to work or family-related constraints [5], [38].

When switching to an online environment, researchers have also considered the broader implications to pedagogies for both programming and in general [8]. One author suggests that an online environment could enhance students' skills such as collaborative problem solving and the capacity to integrate into new communities [8], and the same author considered the challenges of doing group work in synchronous online programming classes, developing a set of recommendations for instructors wishing to include group work in virtual classrooms [9].

3.1 MOOCs in Introductory Programming

MOOCs in programming are discussed as a separate subsection due to their inherent issue in scale and population. MOOCs in general are offered to anyone willing to participate, increasing the amount of students, and hence, creating additional challenges on making support resources available to participants. When comparing published research on MOOCs in introductory programming to the studies on teaching programming online, the amount of MOOC articles is – not surprisingly – lower. Searching for keywords "MOOC" and "programming" in ACM Digital Library produced 58 hits. Out of those articles, 14 include keywords "MOOC", "programming", "CS1", and only a few actually discuss MOOCs in introductory programming. Some of the articles discuss overall experiences in creating more advanced MOOCs (e.g. courses on software engineering) and mention introductory programming (see e.g. [18]), while some discuss possible challenges related to offering MOOCs to specific populations such as K-12 students (see e.g. [35]).

Only a few articles that describe the creation and evaluation of a MOOC in introductory programming exist. One article describes support mechanisms and activities used in a MOOC in programming that was directed to high-school students [65], while a follow-up article discussed how students recruited via the MOOC had fared in their CS studies when compared to other students [66]. While the former studies are somewhat constrained due to the low amount of participants, more recent studies have given insight on larger courses. Recently, a course that focuses on emphasizing human-human interaction in a programming MOOC [71] was

[1] The queries reported in this section have been performed in April 2014

published; the team discusses their platform in a separate article [59]. Another laudable MOOC-effort is the *Functional Programming Principles in Scala*-course, which was offered at Udacity [48].

To date, only one article has evaluated several MOOCs for teaching programming; the result from the evaluation was that MOOCs did not leverage the years of research into teaching programming [4].

3.2 Tools

Nearly any search for information about online programming will produce results about tools. Unfortunately, many of the resulting papers do not address online programming courses, instead focusing on online tools for traditional programming courses. This is not simply a side effect of poor searching techniques, as many of the articles contain the words "online" and/or "programming" in the title.

As an example, consider the top fifteen articles discussing tools returned in a search of the ACM Digital Library performed using the keywords "online programming." Of these, eleven of them discussed systems designed to be used exclusively in on-site programming courses. The remaining four were only tangentially related to online programming courses. One was a proposal for a prototype-only massive multiplayer online role playing game, although the paper made no indication that the game had been fully implemented and deployed and did not suggest that the programming courses that used it would be for anything other than students physically present in a classroom[41]. Two described tools for "blended courses" in which the tool was the only online component[25], [54]. The fourth analyzed how various online assessment systems could be used for a MOOC and argued that the authors' tool was superior in many ways to other assessment systems, although the tool had not been tested in a MOOC[61]. A common direction for recent work on tools is to produce online IDEs, which was also represented in the results of the query[25].

While the creation of tools that enable better student learning is important for both on-site and online programming situations, it appears that the majority of tools with the label online are in fact tools for teaching on-site programming. For more information on tools used in introductory programming, see [49].

3.3 Suggestions from the Literature

The reviewed literature for online programming in many cases contained suggestions and guidelines for creating courses that were not supported by rigorous study. This is a common situation for experience reports, which serve as guide into new directions for educators. However, when considering the work on online programming education the suggestions one finds seem obvious. For example, the recommendations in the online programming literature include:

- Learn to utilize the capabilities of web; make the most of videos, audio and interactive content to engage multiple sensors; identify appropriate tools [70], [72], [36], [60], [21]
- Foster regular interaction and collaboration among the students to help them stay engaged; if needed, use a LMS or some other tool to provide facilities for discussion [9], [70], [51], [21]
- Ask and provide feedback; feedback can be used to both support students in their learning and to improve the course content [21]; respond to student inquiries in a timely fashion [60]
- Design the learning tasks to be informative and make the content incremental; if possible, use a tool that provides runtime syntax and error checking [72], [65], [38], [9]

These recommendations, which are common sense for experienced educators, apply equally well to both online and on-site courses. More importantly, they provide a strong contrast with the recommendations for teaching on-site programming, which are based on theories about learning and/or have been empirically tested for validity. This is an indicator of the relative immaturity of the online programming literature.

4. Discussion of Unresolved issues

Experience reports often serve a good purpose, for example by demonstrating that an approach or idea has merit for further consideration. The next step to take is a more rigorous investigation of pedagogies, tools, and learning environments as well as learner perspectives, which seek to provide answers and information for adoption. In this paper, we turned to the extensive research on teaching programming in traditional contexts for direction. Its offering of teaching practices, derived from instructional theory and based on empirical study, provides useful goals for the less developed study of teaching programming online.

Teaching programming online is an area where experience reports still heavily predominate the literature, indicating that the research still remains in the early stages. Part of the reason may be that many educators have their first teaching experiences in a face-to-face environment. This makes it natural to replicate that experience when creating online courses, which requires using tools to produce the types of interaction seen in the on-site classroom. While approaching teaching programming online as a translation process from existing face-to-face experiences is natural, it may be also limited by personal experiences and things that we do not know [26]. Moving forward from replicating in-person experiences might help online programming classes to take advantage of the unique strengths of the online environment.

The evolution of online programming as a research area would involve investigating many of the questions that have been studied for on-site programming courses, including:

- How can interaction, whether student-student, student-instructor, or student-content, be improved?
- How can tools be used to improve interaction without unduly increasing cognitive load? Comparative studies are especially useful for this question.
- How does learning programming in an online environment affect student course outcomes in later courses and projects?
- How effective are various pedagogies, tools, and techniques? Here empirical studies that can be replicated are a necessity.
- What approaches can be shown to work in comparative studies, whether across student populations, institutions, or cultures?

Note that these points call for studies that evaluate the effectiveness of practices and their online counterparts. While previous studies have evaluated online learning at the course-level (see e.g. [38], [23]), practice-level studies will not only provide explanations for why one course delivery is better than another, it will provide directions for further improvement for diverse modes of delivery.

While investigating existing research questions is useful for the online context, there are some questions that are unique to online programming courses such as:

- What is the relationship between the scale of an online programming class and the amount and type of interaction found in that class?
- How do technological advances change the utility of the results in papers related to teaching programming online? What could be done to better withstand the influence of time and technological change?
- What impact does the audience for an online programming class have on the approaches used? Heterogeneous populations, particularly for massive online classes, complicate research studies.

Advancing the body of knowledge in this area requires both consideration of existing work as well as creative out-of-the-box thinking for the design of online programming classes. There is nothing wrong with finding motivation from the latest tools and technology, provided that the resulting course activities produce the desired learning objectives. But a first step in reaching a learning objective is to define one, and this is an area where researchers in online programming have much left to do.

References

[1] T. Anderson. Getting the mix right again: An updated and theoretical rationale for interaction. *The International Review of Research in Open and Distance Learning*, 4(2), 2003.

[2] O. Astrachan and D. Reed. Aaa and cs 1: the applied apprenticeship approach to cs 1. In *ACM SIGCSE Bulletin*, volume 27, pages 1–5. ACM, 1995.

[3] L. J. Barker, K. Garvin-Doxas, and E. Roberts. What can computer science learn from a fine arts approach to teaching? *SIGCSE Bull.*, 37(1):421–425, Feb. 2005.

[4] M. Ben-Ari. MOOCs on introductory programming: A travelogue. *ACM Inroads*, 4(2):58–61, June 2013.

[5] K. Benda, A. Bruckman, and M. Guzdial. When life and learning do not fit: Challenges of workload and communication in introductory computer science online. *Trans. Comput. Educ.*, 12(4):15:1–15:38, Nov. 2012.

[6] J. Bennedsen and M. E. Caspersen. Exposing the programming process. In *Reflections on the Teaching of Programming*, pages 6–16. Springer, 2008.

[7] C. C. Bonwell and J. A. Eison. *Active learning: Creating excitement in the classroom.* School of Education and Human Development, George Washington University Washington, DC, 1991.

[8] M. Bower. Virtual classroom pedagogy. *SIGCSE Bull.*, 38(1):148–152, Mar. 2006.

[9] M. Bower. Groupwork activities in synchronous online classroom spaces. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '07, pages 91–95, New York, NY, USA, 2007. ACM.

[10] K. E. Boyer, R. S. Dwight, C. S. Miller, C. D. Raubenheimer, M. F. Stallmann, and M. A. Vouk. A case for smaller class size with integrated lab for introductory computer science. *SIGCSE Bull.*, 39(1):341–345, Mar. 2007.

[11] G. Braught, T. Wahls, and L. M. Eby. The case for pair programming in the computer science classroom. *Trans. Comput. Educ.*, 11(1):2:1–2:21, Feb. 2011.

[12] P. Brusilovsky, E. Calabrese, J. Hvorecky, A. Kouchnirenko, and P. Miller. Mini-languages: a way to learn programming principles. *Education and Information Technologies*, 2(1):65–83, 1997.

[13] A. S. Carter and C. D. Hundhausen. A review of studio-based learning in computer science. *J. Comput. Sci. Coll.*, 27(1):105–111, Oct. 2011.

[14] M. E. Caspersen and J. Bennedsen. Instructional design of a programming course: a learning theoretic approach. In *Proceedings of the third intl workshop on Computing education research*, pages 111–122. ACM, 2007.

[15] M. Clancy, N. Titterton, C. Ryan, J. Slotta, and M. Linn. New roles for students, instructors, and computers in a lab-based introductory programming course. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '03, pages 132–136, New York, NY, USA, 2003. ACM.

[16] A. Collins, J. S. Brown, and A. Holum. Cognitive apprenticeship: making thinking visible. *American Educator*, 6:38–46, 1991.

[17] S. Cooper, W. Dann, and R. Pausch. Using animated 3d graphics to prepare novices for cs1. *Computer Science Education*, 13(1):3–30, 2003.

[18] S. Cooper and M. Sahami. Reflections on stanford's moocs. *Commun. ACM*, 56(2):28–30, Feb. 2013.

[19] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69:970, 2001.

[20] P. E. Dickson. Using undergraduate teaching assistants in a small college environment. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 75–80. ACM, 2011.

[21] E. M. El-Sheikh. Techniques for engaging students in an online computer programming course. *Journal of Systemics, Cybernetics & Informatics*, 7(1), 2009.

[22] A. Ellis, D. Hagan, J. Sheard, J. Lowder, W. Doube, A. Carbone, J. Robinson, and S. Tucker. A collaborative strategy for developing shared java teaching resources to support first year programming. In *Proceedings of the 4th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '99, pages 84–87, New York, NY, USA, 1999. ACM.

[23] W. Farag. Comparing achievement of intended learning outcomes in online programming classes with blended offerings. In *Proceedings of the 13th Annual Conf. on Information Technology Education*, SIGITE '12, pages 25–30, New York, NY, USA, 2012. ACM.

[24] A. Forte and M. Guzdial. Computers for communication, not calculation: Media as a motivation and context for learning. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii Intl. Conf. on*, pages 10–pp. IEEE, 2004.

[25] S. Friese. Measuring of and reacting to learners' progress in logic programming courses. In *Proceedings of the Fifteenth Annual Confer-*

ence on Innovation and Technology in Computer Science Education, ITiCSE '10, pages 152–154, New York, NY, USA, 2010. ACM.

[26] J. Gal-Ezer and D. Harel. What (else) should cs educators know? *Commun. ACM*, 41(9):77–84, Sept. 1998.

[27] M. Guzdial. A media computation course for non-majors. *SIGCSE Bull.*, 35(3):104–108, June 2003.

[28] M. Guzdial. Exploring hypotheses about media computation. In *Proceedings of the Ninth Annual Intl. ACM Conf. on Intl. Computing Education Research*, ICER '13, pages 19–26, New York, NY, USA, 2013. ACM.

[29] J. Hamer, H. C. Purchase, P. Denny, and A. Luxton-Reilly. Quality of peer assessment in cs1. In *Proceedings of the Fifth Intl. Workshop on Computing Education Research Workshop*, ICER '09, pages 27–36, New York, NY, USA, 2009. ACM.

[30] S. Horwitz, S. H. Rodger, M. Biggers, D. Binkley, C. K. Frantz, D. Gundermann, S. Hambrusch, S. Huss-Lederman, E. Munson, B. Ryder, et al. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. In *ACM SIGCSE Bull.*, volume 41, pages 163–167. ACM, 2009.

[31] C. Hundhausen, A. Agrawal, D. Fairbrother, and M. Trevisan. Integrating pedagogical code reviews into a cs 1 course: An empirical study. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 291–295, New York, NY, USA, 2009. ACM.

[32] S. Huss-Lederman, D. Chinn, and J. Skrentny. Serious fun: peer-led team learning in cs. In *ACM SIGCSE Bulletin*, volume 40, pages 330–331. ACM, 2008.

[33] L. J. White, J. W. Coffey, and E. M. El-Sheikh. Exploring technologies, materials, and methods for an online foundational programming course. *Informatics in Education*, (Vol 7:2):259–276, 2008.

[34] J. Jenkins, E. Brannock, T. Cooper, S. Dekhane, M. Hall, and M. Nguyen. Perspectives on active learning and collaboration: Javawide in the classroom. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, pages 185–190, New York, NY, USA, 2012. ACM.

[35] L. C. Kaczmarczyk. We need to talk. *ACM Inroads*, 5(1):30–31, Mar. 2014.

[36] R. Karsten, S. Kaparthi, and R. M. Roth. Teaching programming via the web: A time-tested methodology. *College Teaching Methods & Styles Journal (CTMS)*, 1(3):73–82, 2005.

[37] P. A. Kirschner, J. Sweller, and R. E. Clark. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41(2):75–86, 2006.

[38] J. Kleinman and E. B. Entin. Comparison of in-class and distance-learning students' performance and attitudes in an introductory computer science course. *J. Comput. Sci. Coll.*, 17(6):206–219, May 2002.

[39] C. M. Lewis, N. Titterton, and M. Clancy. Developing students' self-assessment skills using lab-centric instruction. *J. Comput. Sci. Coll.*, 26(4):173–180, Apr. 2011.

[40] M. C. Linn and M. J. Clancy. The case for case studies of programming problems. *Commun. ACM*, 35(3):121–132, Mar. 1992.

[41] C. Malliarakis, M. Satratzemi, and S. Xinogalos. Towards a new massive multiplayer online role playing game for introductory programming. In *Proceedings of the 6th Balkan Conference in Informatics*, BCI '13, pages 156–163, New York, NY, USA, 2013. ACM.

[42] E. Mazur and R. C. Hilborn. Peer instruction: A user's manual. *Physics Today*, 50(4):68–69, 1997.

[43] B. Means, Y. Toyama, R. Murphy, M. Bakia, and K. Jones. Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies. 2010.

[44] C. S. Miller and A. Settle. When practice doesn't make perfect: Effects of task goals on learning computing concepts. *ACM Transactions on Computing Education (TOCE)*, 11(4):22, 2011.

[45] M. G. Moore. Editorial: Three types of interaction. *American Journal of Distance Education*, 3(2):1–7, 1989.

[46] C. Murphy, R. Powell, K. Parton, and A. Cannon. Lessons learned from a pltl-cs program. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, pages 207–212, New York, NY, USA, 2011. ACM.

[47] N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik. Improving the cs1 experience with pair programming. In *ACM SIGCSE Bulletin*, volume 35, pages 359–362. ACM, 2003.

[48] M. Odersky, L. Rytz, H. Miller, and P. Haller. Functional programming for all! scaling a mooc for students and professionals alike. In *36th International Conference on Software Engineering (ICSE" 14) SEET Track*, number EPFL-CONF-190022, 2014.

[49] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson. A survey of literature on the teaching of introductory programming. In *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*, ITiCSE-WGR '07, pages 204–223, New York, NY, USA, 2007. ACM.

[50] L. Porter and B. Simon. Retaining nearly one-third more majors with a trio of instructional best practices in cs1. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 165–170, New York, NY, USA, 2013. ACM.

[51] T. Reeves, P. Baxter, and C. Jordan. Teaching computing courses - computer literacy, business microcomputer applications, and introduction to programming online utilizing webct. *J. Comput. Sci. Coll.*, 18(1):290–300, Oct. 2002.

[52] K. Renaud, J. Barrow, and P. le Roux. Teaching programming from a distance: Problems and a proposed solution. *SIGCSE Bull.*, 33(4):39–42, Dec. 2001.

[53] M. Rizvi and T. Humphries. A scratch-based cs0 course for at-risk computer science majors. In *Proceedings of the 2012 IEEE Frontiers in Education Conference (FIE)*, FIE '12, pages 1–5, Washington, DC, USA, 2012. IEEE Computer Society.

[54] G. H. B. Roberts and J. L. M. Verbyla. An online programming assessment tool. In *Proceedings of the Fifth Australasian Conference on Computing Education - Volume 20*, ACE '03, pages 69–75, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[55] A. Robins, J. Rountree, and N. Rountree. Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–172, 2003.

[56] B. Simon, M. Kohanfars, J. Lee, K. Tamayo, and Q. Cutts. Experience report: Peer instruction in introductory computing. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 341–345. ACM, 2010.

[57] S. Smith and S. Stoecklin. What we can learn from extreme programming. *J. Comput. Sci. Coll.*, 17(2):144–151, Dec. 2001.

[58] C. Stewart-Gardiner. Improving the student success and retention of under achiever students in introductory computer science. *J. Comput. Sci. Coll.*, 26(6):16–22, June 2011.

[59] T. Tang, S. Rixner, and J. Warren. An environment for learning interactive programming. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 671–676, New York, NY, USA, 2014. ACM.

[60] R. Thomas. Experiences teaching c++ programming online. In *Proceedings of the Fifth Annual CCSC Northeastern Conference on The Journal of Computing in Small Colleges*, CCSC '00, pages 211–219, USA, 2000. Consortium for Computing Sciences in Colleges.

[61] N. Tillmann, J. De Halleux, T. Xie, S. Gulwani, and J. Bishop. Teaching and learning programming and software engineering via interactive gaming. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 1117–1126, Piscataway, NJ, USA, 2013. IEEE Press.

[62] N. Titterton and M. J. Clancy. Adding some lab time is good, adding more must be better: the benefits and barriers to lab-centric courses. In *FECS*, pages 363–367, 2007.

[63] N. Titterton, C. M. Lewis, and M. J. Clancy. Experiences with lab-centric instruction. *Computer Science Education*, 20(2):79–102, 2010.

[64] M. Vesisenaho, M. Duveskog, E. Laisser, and E. Sutinen. Designing a contextualized programming course in a tanzanian university. In *Frontiers in Education Conference, 36th Annual*, pages 1–6. IEEE, 2006.

[65] A. Vihavainen, M. Luukkainen, and J. Kurhila. Multi-faceted support for mooc in programming. In *Proceedings of the 13th annual conference on Information technology education*, pages 171–176. ACM, 2012.

[66] A. Vihavainen, M. Luukkainen, and J. Kurhila. Mooc as semester-long entrance exam. In *Proceedings of the 14th Annual ACM SIGITE*

Conference on Information Technology Education, SIGITE '13, pages 177–182, New York, NY, USA, 2013. ACM.

[67] A. Vihavainen, M. Paksula, and M. Luukkainen. Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 93–98. ACM, 2011.

[68] A. Vihavainen, T. Vikberg, M. Luukkainen, and J. Kurhila. Massive increase in eager tas: Experiences from extreme apprenticeship-based cs1. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 123–128, New York, NY, USA, 2013. ACM.

[69] H. M. Walker. Collaborative learning: A case study for cs1 at grinnell college and austin. *SIGCSE Bull.*, 29(1):209–213, Mar. 1997.

[70] W. Wang. Teaching programming online. In *International Conference on the Future of Education*, 2011.

[71] J. Warren, S. Rixner, J. Greiner, and S. Wong. Facilitating human interaction in an online programming course. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 665–670, New York, NY, USA, 2014. ACM.

[72] J. L. Zachary and P. A. Jensen. Exploiting value-added content in an online course: Introducing programming concepts via html and javascript. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '03, pages 396–400, New York, NY, USA, 2003. ACM.

Creating an Automated Learning Management Tool in an Engineering Course Based on Interdisciplinary Metrics

Quoc-Viet Dang [1] **and Daniel D. Gajski** [1]
[1] Henry Samueli School of Engineering, University of California, Irvine, CA, United States

Abstract - *Online courses have been debated among the UC regents, numerous college administrators, faculty, and students. On one hand, online education can reduce overhead while enrolling more students. Administration wants to enroll more students while incurring minimal overhead costs. However, simply translating the classroom lectures and materials to an online environment does not necessarily produce equivalent student performance and satisfaction from the course compared to an in-class environment. Since there is no standard in online education, it has produced erratic results in terms of student performance and costs to students. A change in teaching and learning methodology by both teachers and students is required to get the most from today's technology and apply it toward improving student performance and participation. This paper presents a management tool and methodology to provide a more individualized learning experience for students in large hybrid and online courses while keeping overall costs and time commitment down as well as improve overall student performance.*

Keywords: *Computer and Web based instruction, Hybrid Methodology, Massive Online Course Management, Computing Education Research, Online Learning Efficacy*

1 Introduction

Public education is in a crisis. The University of California has increased admissions from 50, 291 in Fall 2003 to 82,850 for Fall 2013 [1] [2]. Student-faculty ratios and number of credit hours per faculty member have increased 10% in the last six years alone amidst UC budget cuts [3]. These increases present a problem for running large classes.

Online courses have been proposed and debated among the UC regents, numerous college administrators, faculty, and students. On one hand, online education can reduce overhead while enrolling more students. Administration wants to enroll more students while incurring minimal overhead costs. However, simply translating the classroom lectures and materials to an online environment does not necessarily produce equivalent student performance and satisfaction from the course compared to an in-class environment. Since there is no standard in online education, it has produced erratic results in terms of student performance and costs to students as well as administration. A change in teaching and learning methodology by both teachers and students is required to take the most advantage from today's technology and apply it toward improving student performance and participation. This paper presents a management tool and methodology to provide a more individualized learning experience for students in large hybrid and online courses while keeping overall costs and time commitment down as well as improve overall student performance.

Massive Open Online Courses (MOOCs), are still teaching in a scaled up version of the typical in-class methodology. This methodology in its present state does not work well for larger courses. Translating this method into an online environment makes the situation even worse. Although preliminary analysis [4] shows that the top 1/3 of students tends to still do well with an in-class methodology taught in an online environment, the other 2/3's of students do not perform as well when exposed to the same situation; in fact, the bottom 1/3 of students did much worse relative to the top 1/3 of students.

Different students respond to and perform differently when exposed to different teaching and learning styles [5][6]. Utilizing the results from this initial research and combining it with other research theories and methodologies, a specialized design learning tool can be developed to facilitate and automate the changes required to allow instructors to manage massive online courses significantly better than current available options.

2 Hybrid Overview

Since the hybrid methodology (Figure 1) combines ideas from both in-class and online methods, we will highlight the proposed purposes of the learning tool in this methodology; therefore, the tool and continuing research efforts will be used to address the scalability and overhead issues found in a hybrid course when trying to maintain performance and motivation with increased enrollment. These issues are also present in in-class and online environments, which allows the tool to be used across different methodologies.

One particular basis the hybrid methodology uses to help reinforce students' learning experience and keep them motivated to keep up with the course is the concept of the "Hawthorne effect," in which subjects were more productive

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

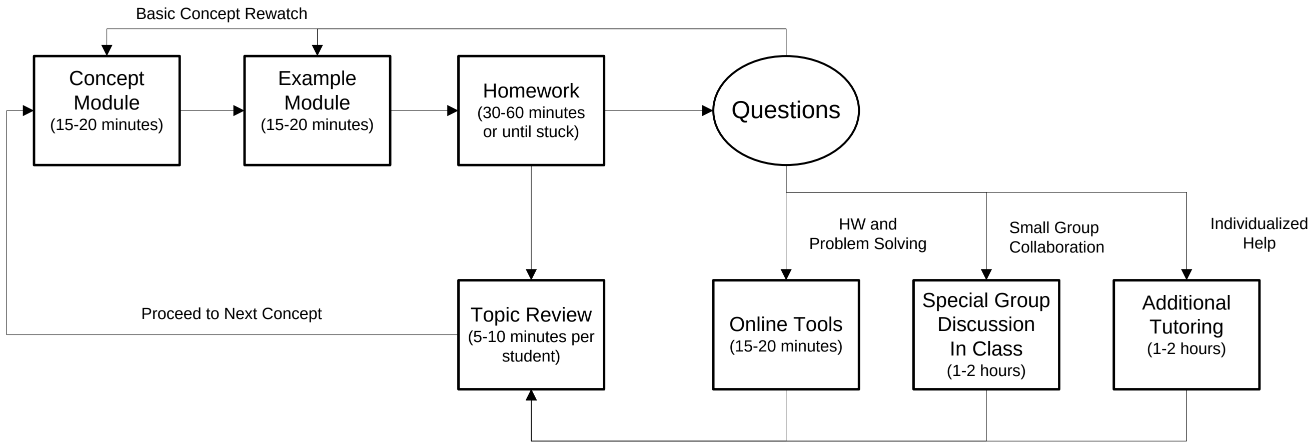

Basic Concept Rewatch
Concept Module (15-20 minutes)
Example Module (15-20 minutes)
Homework (30-60 minutes or until stuck)
Questions
HW and Problem Solving
Small Group Collaboration
Individualized Help
Proceed to Next Concept
Topic Review (5-10 minutes per student)
Online Tools (15-20 minutes)
Special Group Discussion In Class (1-2 hours)
Additional Tutoring (1-2 hours)

Figure 1: The Hybrid Methodology Work Flow.

when they had a manager watching them work [7]. It would not be practical to physically observe individual students did their work, like in Roethlisberger's original findings of worker management at Hawthorne Works in the 1930's [8], especially in a large distance learning course. The proposed methodology including topic review modules and feedback after each video module's questions are answered will be used to satisfy this need. This type of milestone review can guide students and provide feedback on their current weaknesses and strengths and can be automated to be done by a learning management tool.

3 Learning Tool Metrics

To address the metrics measuring student progress based on the varying skill levels of individual students, each module of the design tool will be developed toward fulfilling each of the metrics for each level of students by analyzing how each module fulfills each groups' higher and lower order needs. These needs are adapted from Maslow's Hierarchy of Needs Theory (Figure 2), which identifies a pyramid of needs [9][10][11].

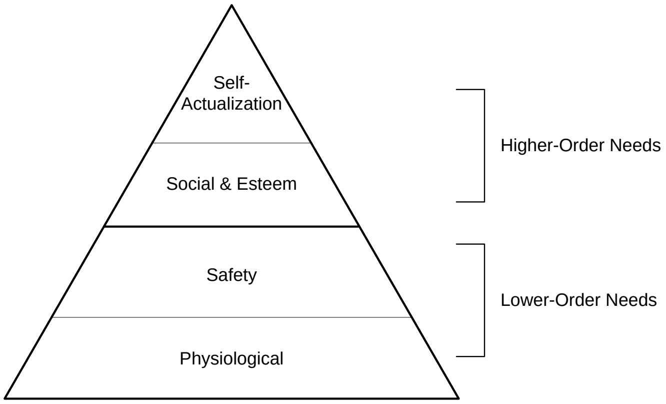


Figure 2: Maslow's Need Hierarchy Model

3.1 Lower-order Needs

Physiological and Safety Needs are lower-order basic needs. In terms of Engineering students, this includes having convenient access to course content and related resources. Furthermore, students need to be able to know their current grade standing throughout the offering of the course. Inconsistent release of material and grades affect their basic needs and limit their motivation, performance, and satisfaction with the course. An intelligent design tool can focus on these lower-order needs by providing ease of uploading and managing of course content for developers & present material to students in a timely and organized manner.

These needs are important for all levels of students, but are especially important for the beginner and intermediate students. By making sure these needs are met, student motivation and satisfaction can be maintained and even increased, allowing the rest of the design tool modules to improve student performance.

3.2 Higher-order Needs

Social and Esteem Needs are higher-order needs. Students need interaction, a feeling of inclusion, and group work. In terms of esteem, students need to build and maintain confidence and feel senses of achievement throughout the course, as well as get recognition when they perform well.

The design tool will provide an interface for developers to set goals and keep track of student achievements. The tool will also provide an area for students to interact with each other. Feedback can be given to individual students on an automated basis to minimize the amount of time developers have to invest in the tool and course to keep these needs satisfied.

The highest level of need, self-actualization, mainly affects advanced students. Students want to fully utilize their skills and abilities with challenging problems and grow as

Engineers. Optional modules and additional advanced material can be customized by developers in the design tool to facilitate this need for students who need more challenge.

4 Learning Tool Overview

The design tool will mainly focus on improving the following metrics: student performance, motivation, and course satisfaction. Students are split into several groups based on their skill and performance level. This tool has a front-end for students and a back-end for instructors and teaching assistants. This design methodology is unique to our design tool and is not currently implemented by other tools managing MOOC's.

instructors create videos and provide questions for concept checking.

Students are presented with a list of concept and example modules. A brief quiz is provided at the end of each module. Students answer these questions and depending upon their performance, the design tool will analyze their answers and make suggestions for additional studying.

For example, a student may have just finished viewing the Arithmetic concept module (**Figure 3**). At this point, some review questions will be prompted to the student, such as:
For a 4-bit Two's Complement Adder/Subtractor, what is the result of 0010 - 1101? ____

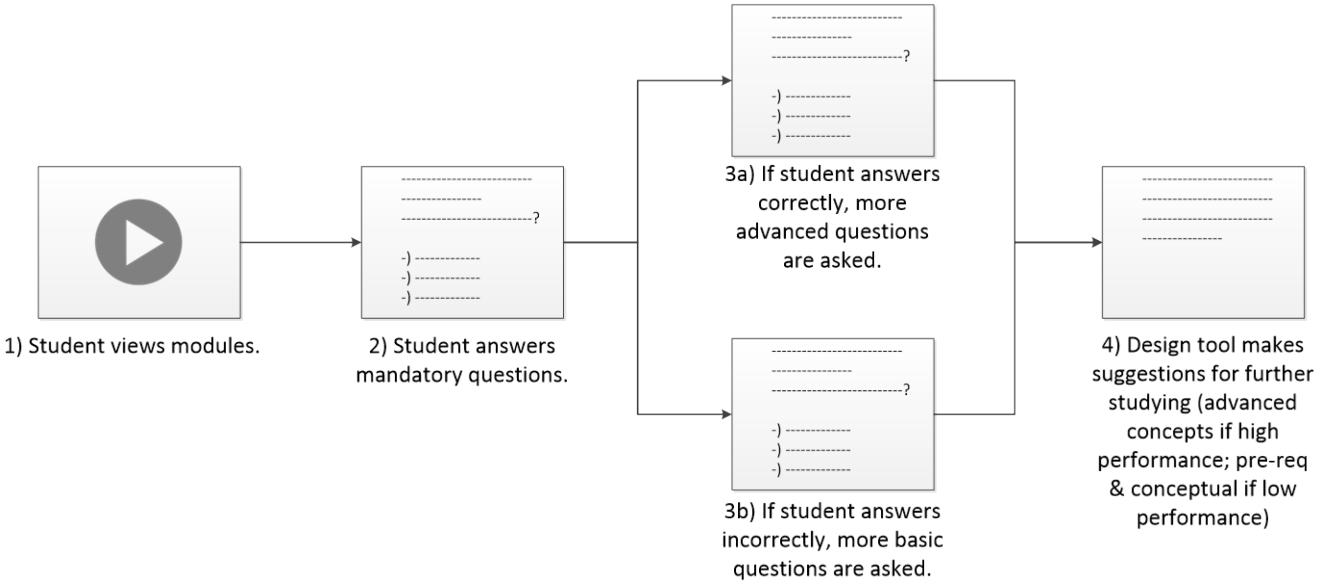

1) Student views modules.
2) Student answers mandatory questions.
3a) If student answers correctly, more advanced questions are asked.
3b) If student answers incorrectly, more basic questions are asked.
4) Design tool makes suggestions for further studying (advanced concepts if high performance; pre-req & conceptual if low performance)

Figure 3: Concept Module Flow

4.1 Front-End Interface

The front-end interface contains modules and tools for students. Each of these modules has a corresponding interface to allow instructors to populate each module with relevant data for the design tool to generate a student profile, analyze the profile, and make suggestions to students. These modules help satisfy basic student needs of the course under Maslow's Hierarchy of Needs Theory. For each section below, the examples provided will be various student performance scenarios for a Digital Design 101 course where the student is progressing through the RTL Combinatorial Components portion of the course.

4.1.1 Concept and Example Modules

Concept and example modules replace the primary lecture and discussion sessions found in traditional classrooms. Each concept and example module is approximately 5-20 minutes in length. Students watch videos and answer questions while

If the student answers correctly with 0101, a more advanced question can be asked to figure out the understanding of the student. If the student answers incorrectly with a wrong answer, more basic questions can be asked to assess the student's level of understanding. The design tool will then compare the keywords associated with the questions against the concept, example, and pre-requisite modules. Based on the answers, the tool suggests further studying material. For example, if the student answered fundamental questions incorrectly, it may determine that they need to review basic binary arithmetic before coming back to two's complement numbers. If the student answered all questions correctly, it may be suggested that the student review more advanced concepts related to adders/subtractors to keep them challenged.

4.1.2 Online Tools

Online tools such as message boards provide an environment for students and instructors to interact and discuss varying concepts regarding the course but are limited

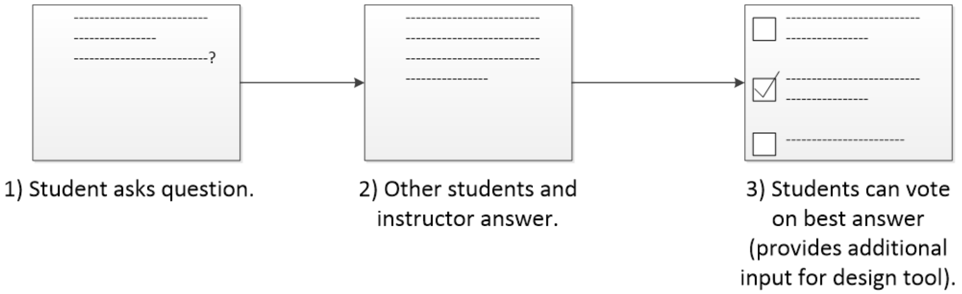

1) Student asks question.
2) Other students and instructor answer.
3) Students can vote on best answer (provides additional input for design tool).

Figure 4: Online Tools Flow

in efficacy. Improving these tools by allowing for more interaction can help satisfy the social and esteem needs portion of Maslow's Hierarchy.

Threads and posts from the course message board can be added to the data set for each student profile by the design tool with help from instructors and teaching assistants (**Figure 4**). Keywords can get generated for each post by the design tool to further analyze the needs of students.

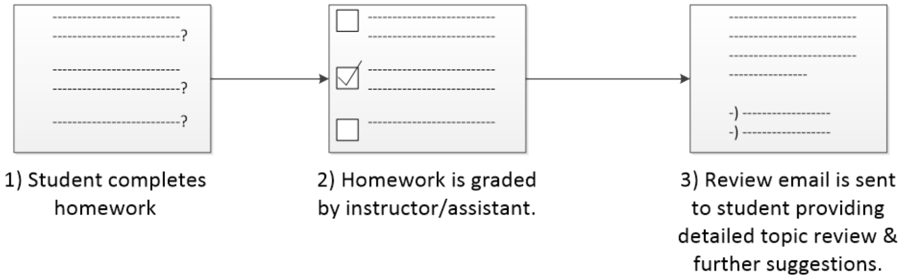

1) Student completes homework
2) Homework is graded by instructor/assistant.
3) Review email is sent to student providing detailed topic review & further suggestions.

Figure 5: Topic Review Flow

The general format will ask a student to classify their own post among several different options, such as "question", "clarification requested", "typo/error found", etc. Posters can also vote for the "best solution" when a question is posed as well.

Continuing with the RTL combinatorial module example with two's complement adders, a student may post a question regarding how to handle overflow in such cases. Other students, and possibly the assistants and instructor, will answer the post. Participants vote on the best answer. The current best answer is highlighted for the topic thread and is later used by the design tool when making studying suggestions.

4.1.3 Topic Review

When homework and initial viewing of the concept and example modules are completed, the topic review can be automatically generated by the design tool, which will send suggestions for further studying to students based on their profile and data gathered from them for each topic. This automated topic review can be sent via email automatically, minimizes the need for instructor or assistant initial intervention, and maintains the students desire for "one-on-one" interaction.

Students receive their review via email after completing the required modules and homework. Homework is downloaded, completed, and submitted online through the design tool or equivalent online interface (**Figure 5**). If homework is submitted online and in the proper format, an intelligent design tool can provide all the grading and post assessment of work. The email will contain information regarding the student's current progresses, his or her strengths and weaknesses, as well as suggestions for additional review to help the student catch up if needed. For advanced students, additional challenging modules are recommended.

The homework for RTL combinatorial components contains several concept questions. Each student's homework is graded by the instructor and assistants. Optional comments may be left for the assignment. The design reviews the homework grades, matches it against previous suggestions, checks student module viewing behavior, and generates a detailed review email to be sent to each student. The review email can suggest more detailed review modules than the suggestions from the concept and example modules.

4.2 Learning Tool Back-End and Algorithms

The design tool contains two main data structures: the course data structure and student profile data structure. Together, both structures are utilized by the design tool to analyze student performance and make appropriate suggestions for students while providing a useful profile to display to the instructor for reference.

The design tool analyzes data from the design tool data structure in order to create custom reports for students and assign them additional modules for further studying when appropriate. This tool also provides a summary of each student's progress for instructors to review before meeting with students individually when needed to maximize efficiency when holding office hours.

4.2.1 Student Feedback Generation

After students each concept and example module, students answer a series of questions to check their understanding of the material. Depending on their answers, the design tool will rate the students, look through the data structure for additional modules that will help them, and provide additional feedback as appropriate.

The design tool can bases its keyword search on a Latent Semantic Analysis (LSA) algorithm or other natural language processing algorithm to find similar keywords and concepts based on the provided dataset. The design tool can index message board posts and reference the data inputted for each post by instructors and assistants, allowing for more information to be analyzed, increasing potential custom guidance for each student.

The LSA algorithm is used to find modules, posts, and questions based on the keywords of the quiz or homework questions the student misses. From there, the design tool determines which modules are most appropriate based on difficulty, closeness in relation, and previous topic review suggestion results.

4.2.2 Student Summary for Instructor Generation

Instructors will inevitably be required to meet with some students, be it in-person or online. Since the goal of the design tool is to allow for an individualized experience for students on a large scale, student summaries are generated by the design tool to help instructors quickly identify a student's weaknesses and strengths before meeting with him or her.

Similar to the Student Feedback Generation, a Latent Semantic Analysis algorithm to find matches. The LSA algorithm is used to find modules, posts, and questions based on the keywords of the quiz or homework questions the student misses and gets correctly. The design tool further analyzes how the student divides his or her time among various concepts. Suggestions are made based on student participation and focus on the modules.

5 Conclusions

This automated learning management tool can categorize and assess students by their performance levels basing its techniques from other disciplines to maximize not just performance but also motivation and self-satisfaction; existing tools for managing MOOC's follow traditional methodologies and provide the same experience for everyone, hindering motivation, self-satisfaction, and inevitably performance for many students.

Furthermore, the data structure is based on comparing student and course profiles to automate teaching and learning, which allows larger scale instruction while minimizing overhead and time commitment from both students and instructors. The structure provides a useful dataset for the design tool, which uses the data in algorithms that generate meaningful reports containing suggestions and reviews of the student's work based on each individual's progress. This automated system is integral in making a hybrid methodology for large scale courses with minimal instructor involvement possible while focusing on satisfying the metrics proposed.

Further research and data gathering for developing this automated tool is currently underway.

6 References

[1] University of California Office of the President, "UC California feshman admissions for Fall 2003 - University of California Office of the President," 6 November 2004. [Online]. Available: http://www.ucop.edu/news/factsheets/2003/admissions_summary.pdf. [Accessed 26 August 2013].

[2] University of California Office of the President, "UC California freshman admissions for Fall 2013 - University of California Office of the President," 17 April 2013. [Online]. Available: http://www.ucop.edu/news/factsheets/2013/fall_2013_admissions_table2.pdf. [Accessed 26 August 2013].

[3] University of California Academic Senate Source, "Report Details Trends for Faculty Workload," University of California Academic Senate - The Senate Source, June 2013. [Online]. Available: http://senate.universityofcalifornia.edu/news/source/facultyworkload.june2013.html. [Accessed 26 August 2013].

[4] Daniel D. Gajski, Quoc-Viet Dang and WenLiang He, "An Online Methodology for Individualized Education," in 2013 International Conference on e-

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Learning, e-Business, EIS, & e-Government (EEE'13), Las Vegas, NV, July 22-25, 2013.

[5] Omer Delialioglu and Zahide Yildirim, "Design and development of a technology enhanced hybrid instruction based on MOLTA model: Its effectiveness n comparison to traditional instruction.," Computers & Education, vol. 51, no. June 2007, pp. 474-483, 2008.

[6] J.W. Baker, "The 'Classroom Flip': Using Web Course Management Tools to Become the Guide on the Side.," in 11th Conference on College Teaching and Learning, Jacksonville, FL, 2000.

[7] M. Anteby and R. Khurana, "A New Vision - The Human Relations Movement - Baker Library | Bloomberg Center, Historical Collections," 30 June 2007. [Online]. Available: http://www.library.hbs.edu/hc/hawthorne/. [Accessed 21 August 2013].

[8] F. J. Roethlisberger, W. Dickson and H. A. Wright, Management and the Worker: an Account of a Research Program Conducted by the Western Electric Company, Hawthorne Works, Chicago., Cambridge: Harvard University Press, 1939.

[9] S.A. McLeod, "*Maslow's Hierarchy of Needs - Simply Psychology,"* 2007. Retrieved from http://www.simplypsychology.org/maslow.html. Access October 22, 2013.

[10] A.H. Maslow, "A Theory of Human Motivation," 1943. *Psychological Review*, 50(4), 370-96.

[11] R. Villa, J. Thousand , W. Stainback, and S. Stainback, *Restructuring for Caring & Effective Education*. Baltimore: Paul Brookes, 1992.

An Interactive Multimedia Website for Teaching and Learning Engineering Design and Manufacturing

Longhan Xie[1,2] **and Ruxu Du**[1]
[1]Institute of Precision Engineering, Chinese University of Hong Kong, China
[2]School of Mechanical and Automotive Engineering, South China University of Technology, China

Abstract - *This work was aimed at presenting an interactive learning multimedia website for the Engineering Design and Manufacturing courses. This interactive website provides various tools facilitating both classroom teaching and self-learning. There are two kinds of materials in the website: the first kind follows the divide-and-conquer methodology and gives fundamental knowledge in small segments under a clear structure, and the second kind includes a number of practical engineering examples following a systematic line. The website contains rich multimedia material to assist students to master the materials step-by-step. Cases study showed that the arrangement and contents of the website can accelerate the learning of engineering design and manufacturing*.

Keywords: Inactive learning; E-learning; Engineering design and manufacturing.

1 Introduction

Teaching Engineering Design and Manufacturing is a challenging task. Presently, the usual teaching and learning method is as follows: teachers prepare the presentation and lecture notes, go to class, and deliver the lecture. Students go to class with little preparation, attend lecture, and later do exercises and/or projects on their own. This “push” teaching method has a number of limitations. First, while there are good books, presentations and lecture notes available, the students may not have a complete grasp of what has been taught and what the practical uses are. This is particularly true for engineering design and manufacturing studies because the students have a more limited understanding of how engineering parts are designed and made. Second, because the material is pushed onto the students, they have little time to build up their skillset and experience. They often just copy and regurgitate, rather than truly understanding the material. In addition, engineering design and manufacturing requires a very board knowledge base, including calculus, physics, chemistry, materials, mechanics, heat and fluid, circuit, computer, etc. Drained by many other courses, the “push” method may not be effective.

In recent years, with the advance of computer and network technologies, the active learning method becomes more and more effective. When active learning is compared to traditional teaching methods, students achieve better comprehension, retain the information longer and enjoy the class more [1]–[4]. In fact, a major benefit of the use of active methodologies is that they contribute to students developing the capacity actively to research and undertake responsibility for their own learning process, and to solve problems with their own resources. A successful example is the Khan Academy [5]; under the support of the Bill and Melinda Gates Foundation, it has delivered over 176 million lessons to students around the world. For engineering design and manufacturing, efforts have been made by Bookboon [6] and e-Books [7]. In [8] a set of web-based java applets was introduced for the study of properties of electronic materials. Reference [9] used web-based technology as an interactive learning tool in the theory of computation course. Although these examples provided interactive learning tools, they are not well-organized and do not have multimedia aids. Clearly, developing a multimedia engineering design and manufacturing website is of great importance not only to registered engineering students, but also to engineers around the world. In this paper, we presented our web-based multimedia website, www.dulibrary.org, for interactive learning of engineering design and manufacturing.

2 The Multimedia Website for Interactive Learning

As shown in Figure 1, the website contains two types of materials: the basic knowledge and the tools. Following the divide-and-conquer methodology, the basic knowledge is decomposed into segments under a clear structure. The tools include various computer software packages used in engineering design and manufacturing. In addition, there are many practical engineering examples, by which students can learn step-by-step and challenge themselves to design and make prototypes.

From a structure point of view, the website, as shown in Fig. 1, the website has three parts: Design, Manufacture and Analysis. The Design part consists of four modules: including two basic knowledge modules: TRIZ and GD&T, and two tools module: SOLIDWORKS and AUTOCAD. TRIZ is a theory for conceptual design. GD&T module provides an important tool for assembly of complex products. The two tools modules guide the student to learn the software system

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

step-by-step and function-by-function. They both contain many illustrations and practical engineering examples. In particularly, SOLIDWORKS is better in 3D design while AutoCAD is convenient for 2D drawing. With rich multimedia illustrations and plenty of practical examples, students at any level can learn the skills interactively.

The Manufacture part consists of four modules including three basic knowledge modules (Engineering Materials, CNC, and 3D printing) and a tool module: MASTERCAM. The Engineering Materials module not only describes the basics of engineering materials but also links to the website www.matweb.com, which allows the student to locate specific materials. The CNC (Computerized Numerical Control) module gives the basics of CNC machine tool and its processes. The 3D printing module describes various 3D printing technologies and how to use them to make a prototype quickly. The MASTERCAM module provides the illustrations and engineering examples on how to program CNC machine tools. With these knowledge and tools, students can possess the fundamental skills for making a prototyping product.

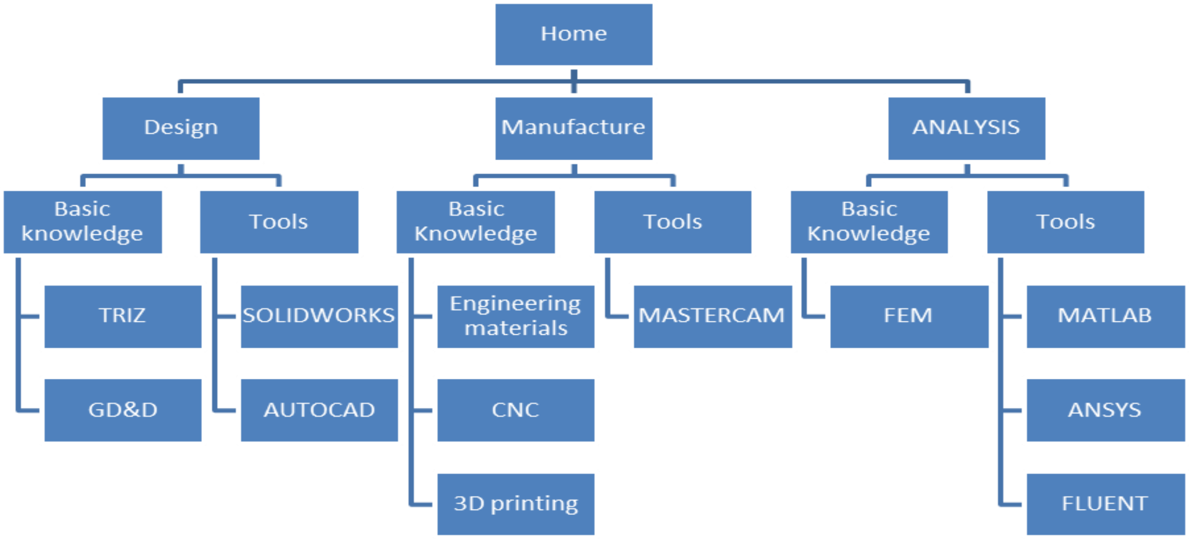
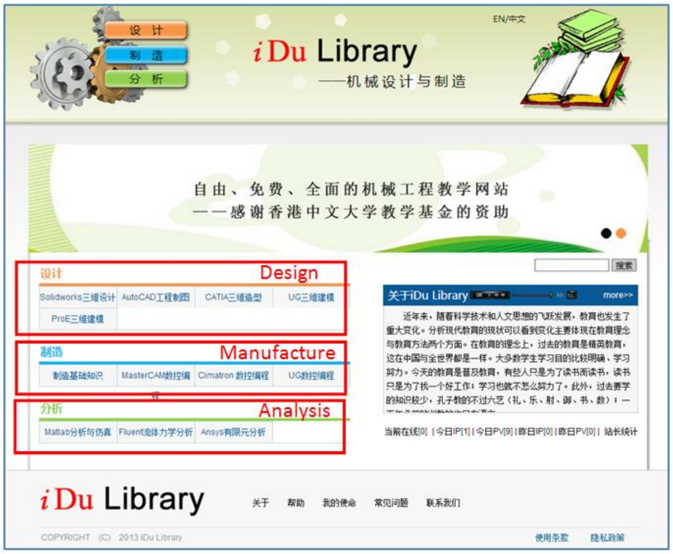


Fig. 1: The framework of the website

The analysis part is includes one basic knowledge module and three tools modules. The FEM (Finite Element Modeling) gives the basic knowledge of the finite element analysis. The three tools module helps to students to learn FEM software packages: MATLAB, ANSYS, and FLUENT. In particularly, ANSYS is used for structure analysis and FLUENT is used for fluid and thermal analysis. MATLAB is used for design analysis and optimization.

The first version of the multimedia website has been launched; the main page is shown in Fig. 2. The concise main page directly shows the aforementioned three parts for engineering design and manufacture, based on which students can study the materials step-by-step or by examples. For example, when a student wants to learn the skills of using SOLIDWORKS, he / she can select the 'SOLIDWORKS' module. The SOLIDWORKS module page is shown in Fig. 3, where a number of lectures are listed in a systematic order, such as sketch a design, model a design, design a component, assembly, and animation. Each lecture may be further decomposed into talks giving further details. For example, the sketch module contains the talks about sketch a line, a curve, a polygonal shape and etc. Fig. 4 shows the talk on making a solid using rotation. This divide-and-conquer structure helps the student to gain the systematic view of the technology and the tools.



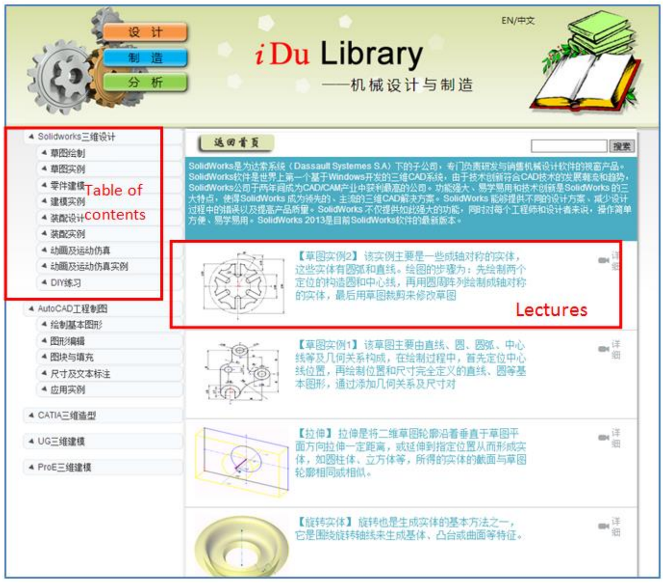
自由、免费、全面的机械工程教学网站
——感谢香港中文大学教学基金的资助

近年来，随着科学技术和人文思想的飞跃发展，教育也发生了重大变化。分析现代教育的现状可以看到变化主要体现在教育理念与教育方法两个方面。在教育的理念上，过去的教育是精英教育，这在中国与全世界都是一样。大多数学生学习目的比较明确、学习努力。今天的教育是普及教育，有些人只是为了读书而读书，读书只是为了找一个好工作！学习也就不怎么努力了。此外，过去要学的知识较少，孔子教的不过六艺（礼、乐、射、御、书、数）；一

当前在线[0] | 今日IP[1] | 今日PV[9] | 昨日IP[0] | 昨日PV[0] | 站长统计

COPYRIGHT (C) 2013 iDu Library

Fig. 2: The main page of the website



设 计
制 造
分 析

*i*Du Library
——机械设计与制造

EN/中文

- Solidworks三维设计
  - 草图绘制
  - 草图实例
  - 零件建模
  - 建模实例
  - 装配设计
  - 装配实例
  - 动画及运动仿真
  - 动画及运动仿真实例
  - DIY练习
- AutoCAD工程制图
  - 绘制基本图形
  - 图形编辑
  - 图块与填充
  - 尺寸及文本标注
  - 应用实例
- CATIA三维造型
- UG三维建模
- ProE三维建模

返回首页 搜索

SolidWorks是为达索系统（Dassault Systemes S.A）下的子公司，专门负责研发与销售机械设计软件的视窗产品。SolidWorks软件是世界上第一个基于Windows开发的三维CAD系统，由于技术创新符合CAD技术的发展潮流和趋势，SolidWorks公司于两年间成为CAD/CAM产业中获利最高的公司。功能强大、易学易用和技术创新是SolidWorks的三大特点，使得SolidWorks成为领先的、主流的三维CAD解决方案。SolidWorks能够提供不同的设计方案、减少设计过程中的错误以及提高产品质量。SolidWorks不仅提供如此强大的功能，同时对每个工程师和设计者来说，操作简单方便、易学易用。SolidWorks 2013是目前SolidWorks软件的最新版本。

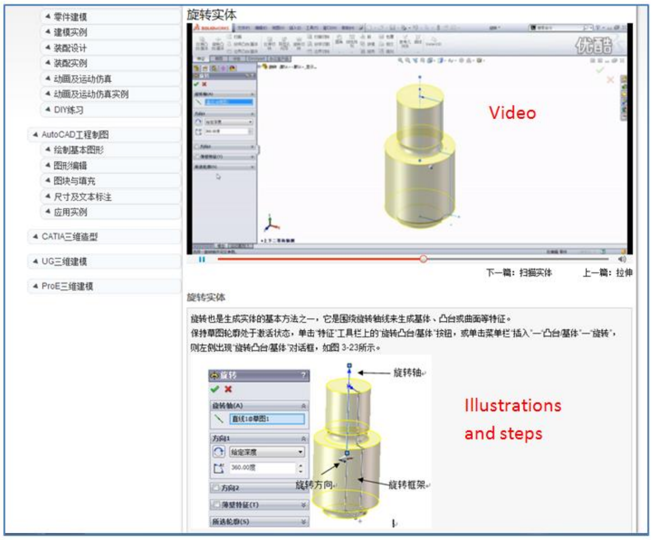
【草图实例2】该实例主要是一些成轴对称的实体，这些实体有圆弧和直线。绘图的步骤为：先绘制两个定位的构造圆和中心线，再用圆周阵列绘制成轴对称的实体，最后用草图裁剪来修改草图 详细

【草图实例1】该草图主要由直线、圆、圆弧、中心线等及几何关系构成，在绘制过程中，首先定位中心线位置，再绘制位置和尺寸完全定义的直线、圆等基本图形，通过添加几何关系及尺寸对 详细

【拉伸】拉伸是将二维草图轮廓沿着垂直于草图平面方向拉伸一定距离，或延伸到指定位置从而形成实体，如圆柱体、立方体等，所得的实体的截面与草图轮廓相同或相似。 详细

【旋转实体】旋转也是生成实体的基本方法之一，它是围绕旋转轴线来生成基体、凸台或曲面等特征。 详细

Fig.3: The SOLIDWORKS module



- 零件建模
- 建模实例
- 装配设计
- 装配实例
- 动画及运动仿真
- 动画及运动仿真实例
- DIY练习
- AutoCAD工程制图
  - 绘制基本图形
  - 图形编辑
  - 图块与填充
  - 尺寸及文本标注
  - 应用实例
- CATIA三维造型
- UG三维建模
- ProE三维建模

旋转实体

下一篇：扫描实体 上一篇：拉伸

旋转实体

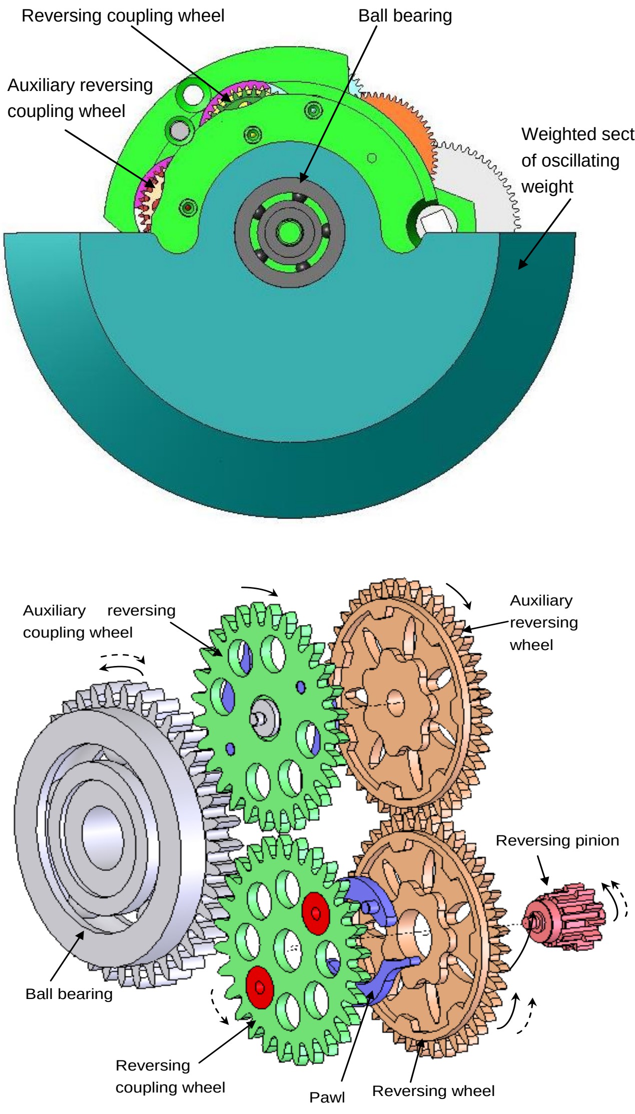
旋转也是生成实体的基本方法之一，它是围绕旋转轴线来生成基体、凸台或曲面等特征。
保持草图轮廓处于激活状态，单击"特征"工具栏上的"旋转凸台/基体"按钮，或单击菜单栏"插入"—"凸台/基体"—"旋转"，则左侧出现"旋转凸台/基体"对话框，如图 3-23所示。

Fig.4: The talk about making a solid using rotation in SOLIDWORKS

On the other hand, students can also learn from practical cases as shown the subsequent section.

3 A case study

In this section, an example is presented to demonstrate how to utilize the website to learn engineering design and analysis. This was a Final Year Project of an undergraduate student, whose main task was to analyze the automatic winding mechanism in a mechanical watch movement. The design is shown in Fig. 5. As a final year student in mechanical engineering, he or she would have theoretical knowledge to do the project, but is typically lack of skills and tools to complete the project. In the project, the student would need to design all the components of the automatic winding mechanism, and then assemble them together. In this case, he / she would first use SOLIDWORKS to model the parts and then do the assembly. Next, he / she would to study the GD&T module to setup the geometric dimensions and tolerances for assembly. Then, he / she would generate the 2D engineering drawing to provide manufacturing information.



Reversing coupling wheel
Ball bearing
Auxiliary reversing coupling wheel
Weighted sector of oscillating weight
Auxiliary reversing coupling wheel
Auxiliary reversing wheel
Reversing pinion
Ball bearing
Reversing coupling wheel
Pawl
Reversing wheel

Fig. 5: The CAD model of the automatic winding device

After completing the design and assembly, the student need to build a mathematical model to analyze the performance of the automatic winding mechanism under working conditions. First, he / she would use the 2D drawing to assist the understanding the relations among components, as shown in Fig. 6. This again can be done using SOLIDWORKS. With the mathematical models and equations, as show in Eq. (1), he / she can analyze the performance of the winding mechanism using MATLAB. Finally, he / she can also plot the simulation results in figures, as shown in Fig. 7.

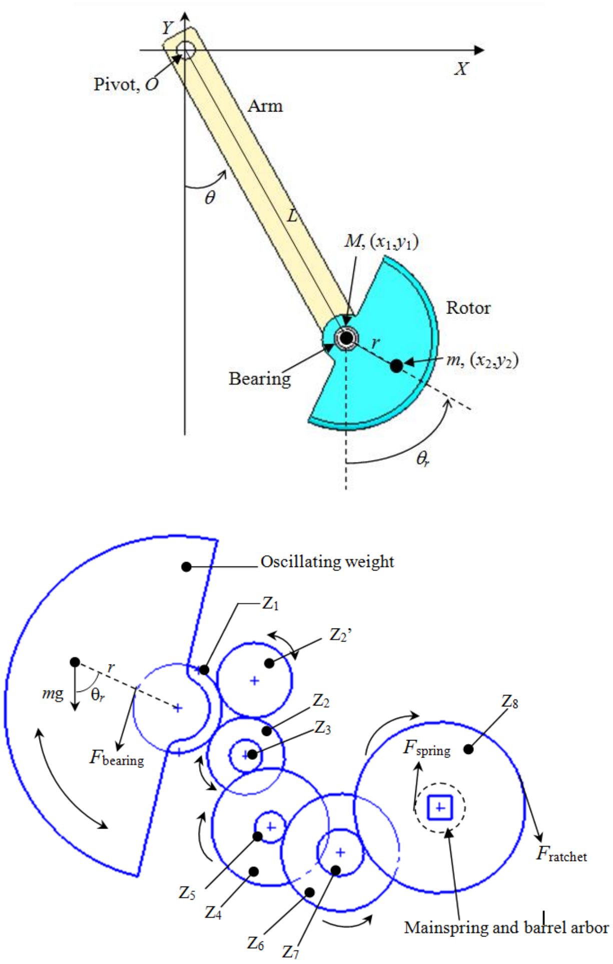


Fig. 6: Schematic diagram of winding system when arm is swinging

$$\begin{cases} \ddot{\theta} = \dfrac{-g(2M+m)\sin\theta - mg\sin(\theta-2\theta_r) - 2\sin(\theta-\theta_r)m(\dot{\theta}_r^2 r + \dot{\theta}^2 L\cos(\theta-\theta_r))}{L(2M+m-m\cos(2\theta-2\theta_r))} \\ \ddot{\theta}_r = \dfrac{2\sin(\theta-\theta_r)(\dot{\theta}^2 L(M+m) + g(M+m)\cos\theta + \dot{\theta}_r^2 rm\cos(\theta-\theta_r))}{r(2M+m-m\cos(2\theta-2\theta_r))} \end{cases} \quad (1)$$

It shall be pointed that in the entire case, the student can study by himself / herself through the multimedia website to master the necessary skills and tools to complete a project in a short period of time.

From this example, it is seen that the multimedia website provides an effective tool for students to quickly learn practical skills without much assistance from professors or tutors.

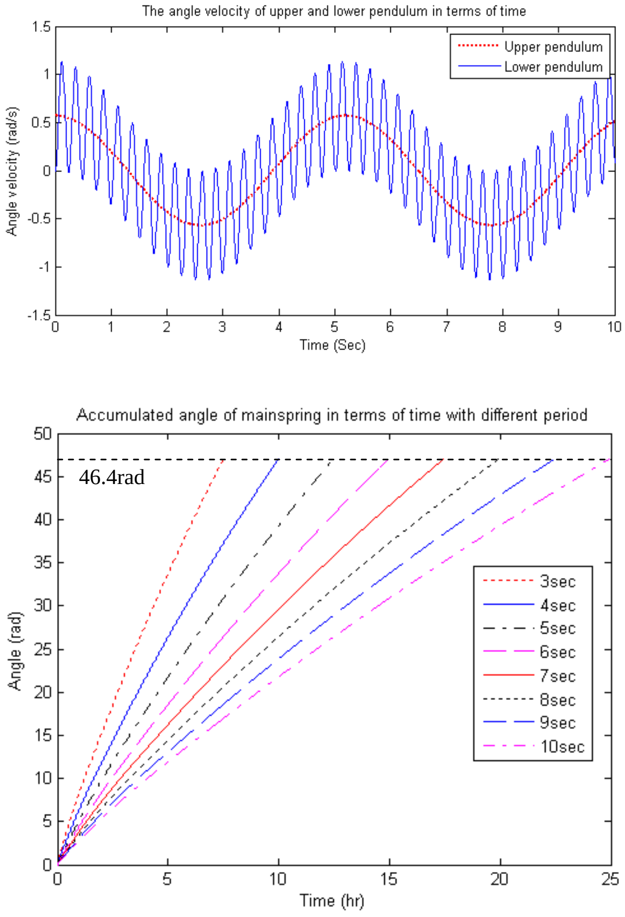


Fig. 7: results for analysis

4 Conclusions

Based on the above discussions, the following conclusions can be made:

(1) A multimedia website was developed to provide an interactive learning tool specifically for the Engineering Design and Manufacturing courses. It is an effective and efficient way to assist learners to quickly master skills and tools on design and manufacturing.

(2) The website contains two kinds of lectures: the basic knowledge lectures and the practical tools lectures. The basic knowledge lectures follow the divide-and-conquer methodology and provides fundamental knowledge in small segments under a clear structure. The tools lectures give students step-by-step practical engineering examples, which can facilitate the learning.

(3) A number of case studies are included, which helps to accelerate the learning of engineering design and manufacturing.

Presently, the website is largely in Chinese. However, an English version will be developed soon.

Acknowledgement

This project was partially supported by the Teaching Development Grants for 2012-15 Triennium of the Chinese University of Hong Kong.

References

[1] R. M. Felder, G. N. Felder, and E. J. Dietz, “A longitudinal study of engineering student performance and retention. V. comparisons with traditionally- taught students,” J. Eng. Educ., vol. 87, no. 4, pp. 469–480, Oct. 1998.

[2] B. Higgs and M. McCarthy, “Active learning—From lecture theatre to field-work,” in merging Issues In The Practice of University Learning and Teaching, G. O’Neill, S. Moore, and B. McMullin, Eds. Dublin, Ireland: AISHE, 2005, pp. 37–44.

[3] M. Prince, “Does active learning work? a review of the research,” J. Eng. Educ., vol. 93, no. 3, pp. 223–231, Jul. 2004.

[4] B. Timmerman and R. Lingard, “Assessment of active learning with upper division computer science students,” in Proc. 33rd ASEE/IEEE Frontiers in Educ. Conf., 2003, pp. S1D–7.

[5] Khan Academy, http://www.khanacademy.org/

[6] Book Boon, www.bookboon.com

[7] E-books Directory, www.e-booksdirectory.com/

[8] J. Masters and T. Madhyastha, “Educational Applets for Active Learning in Properties of Electronic Materials,” IEEE Transactions on Education, Vol. 48, No. 1, 2005.

[9] M. Hamada, “Visual Tools and Examples to Support Active ELearning and Motivation with Performance Evaluation,” Lecture Notes in Computer Science, vol. 3942, pp. 147-155, 2006.

Feeling Glassy:
An Analysis of Developing for Google Glass

Dakota Gallimore
Department of Mathematics and Computer Science
University of Virginia, College at Wise
Wise, VA USA

Joseph LaCava
Department of Mathematics and Computer Science
University of Virginia, College at Wise
Wise, VA USA

Dr. Abrar Qureshi
Department of Mathematics and Computer Science
University of Virginia, College at Wise
Wise, VA USA

Abstract - *Google Glass is considered a front-runner of mobile computing. It functions as a wearable device that, as the name suggests resembles a pair of glasses, but the abilities of the device prove that this is no ordinary pair of spectacles. Understanding that this device stands to be the new and, currently, unrivaled device for the future of mobile computing, those with the ability to program for it stand to enter into a market that has limitless potential. The proposed paper serves as an analysis of the development process and attempts to outline the pieces that are required for both Android and Glass.*

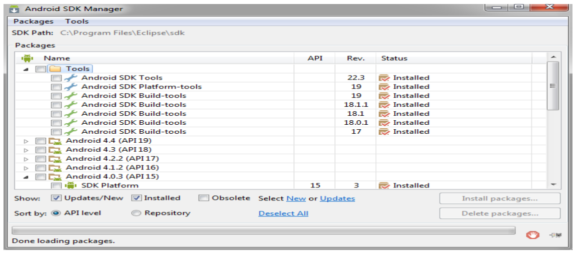
Keywords: Activities, Services, Google Glass, Intents, Mirror API, Google Development Kit (GDK).

1. Introduction

Programming for Google Glass starts as any other programming project would, with the SDK. Google is currently in the stages of developing their own Android Studio, but for the purposes of coding for Glass we, the developers, utilized the existing Eclipse[1] and the ADT plugin bundle. Both, the Android SDK and the Android Studio, are available to download for free on the Google development website. As one might suspect from the IDE, Java[2] is the suggested language used in Android programming. However, for those who prefer C/C++ or simply wish to reuse existing libraries, Google also offers a NDK, but does not suggest its use to program in C/C++ and states using native code does not typically increase application performance. All of these and more is available of Google's developer site. The sample project that is later referenced in this paper was built in the Android SDK environment, the Eclipse IDE and Android ADT plugin bundle. Further detail of how to set up the IDE and importing the appropriate libraries will be covered later in this paper.

2. Project requirements

Assuming that the steps to download the SDK above were followed correctly and everything is working, if issues please consult the support provided by Google's website, it is time to download the appropriate library for GDK. Under Window, in the menu bar, select the Android SDK Manager which will pull up a window similar to the one below.



SDK Path: C:\Program Files\Eclipse\sdk

Packages

Name	API	Rev.	Status
◢ □ Tools			
□ Android SDK Tools		22.3	Installed
□ Android SDK Platform-tools		19	Installed
□ Android SDK Build-tools		19	Installed
□ Android SDK Build-tools		18.1.1	Installed
□ Android SDK Build-tools		18.1	Installed
□ Android SDK Build-tools		18.0.1	Installed
□ Android SDK Build-tools		17	Installed
▷ □ Android 4.4 (API 19)			
▷ □ Android 4.3 (API 18)			
▷ □ Android 4.2.2 (API 17)			
▷ □ Android 4.1.2 (API 16)			
◢ □ Android 4.0.3 (API 15)			
□ SDK Platform	15	3	Installed

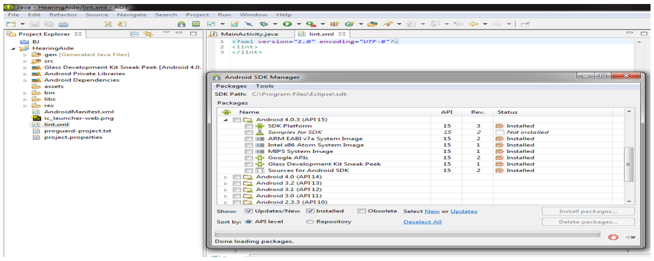
Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

Figure 1. Android SDK Manager.

Make sure the most recent SDK tools have been downloaded and installed. All of the programming for Glass is through the Android 4.0.3 or API15, as is depicted in the screenshot below. You will need to download the entire API, excluding the samples at your prerogative, so it may be a better decision to do all of them at once. It should be noted that in the extras folder resides all of the legal documentation for android development. It may be a good idea to download and install these as well.



```
BJ
◢ HearingAide
  ▷ gen [Generated Java Files]
  ▷ src
  ▷ Glass Development Kit Sneak Peek [Android 4.0.
  ▷ Android Private Libraries
  ▷ Android Dependencies
    assets
  ▷ bin
  ▷ libs
  ▷ res
    AndroidManifest.xml
    ic_launcher-web.png
    lint.xml
    proguard-project.txt
    project.properties
```

```
<?xml version="1.0" encoding="UTF-8"?>
<lint>
</lint>
```

SDK Path: C:\Program Files\Eclipse\sdk

Packages

Name	API	Rev.	Status
◢ □ Android 4.0.3 (API 15)			
□ SDK Platform	15	3	Installed
□ *Samples for SDK*	*15*	*2*	*Not installed*
□ ARM EABI v7a System Image	15	2	Installed
□ Intel x86 Atom System Image	15	1	Installed
□ MIPS System Image	15	1	Installed
□ Google APIs	15	2	Installed
□ Glass Development Kit Sneak Peek	15	1	Installed
□ Sources for Android SDK	15	2	Installed
▷ □ Android 4.0 (API 14)			
▷ □ Android 3.2 (API 13)			
▷ □ Android 3.1 (API 12)			
▷ □ Android 3.0 (API 11)			
▷ □ Android 2.3.3 (API 10)			

Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

Figure 2. Android API 15

The Figure 2 shows the Android API 15 open and as you can see installed. If there any problems installing or locating the correct API bundles please contact Google, or simply refer to their developer website. To continue into the depth of the GDK coding process, we have to give a few descriptions of the parts found and implemented in Android development.

3. Development parts[4]

3.1 Activities

Activities are the main building blocks of any android project. Activities handle anything having to do with user interaction and maintaining the user interface thread. Activities follow a lifecycle process. Developers can use this to contain actions performed within their projects. Such actions could include shutting down a running service for power management or memory optimization and management.

3.2 Services

Services handle background tasks much like AsyncTasks, however it is important to note that AsyncTasks are more suited for short term processes like downloading a file and then updating the UI thread to reflect completion whereas services are fitted for long running background tasks. This could be anything from audio playback to polling a WiFi connection in a set interval.

3.3 Fragments

Fragments are also play a keen role in android development and while Glass is based off android 4.0.3, they do not have a substantial role in Glass development but are worth noting. Fragments are a part of an activity and you can have multiple nested fragments per activity to give multiple viewpoints within an activities UI thread. This means all fragments share the same lifecycle state as their parent activity. Fragments were introduced in android 3.0 as a mean to extend the functionality and dynamic (*dynamic-like*) nature of android activities.

4. Glass environment[5]

The Glass environment is based off a timeline like implementation where events in the past are generated and stored on the left side of the home screen and present or upcoming events are stored. This is where Glass steps away from its fundamental android roots and becomes one in its own. Developers can publish and delete items pertaining to their application onto the Glass timeline. This being said the Glass development standards still apply regardless if the user is interacting with an activity or a timeline card. A time card is an instance of a developer's application being placed on the main timeline. Time cards allow applications to handle many tasks in the background like audio processing.

4.1 Menu Items

Menu items allow users to request actions related to the timeline cards. They are one of the most interesting services included in Glass and allow the user to interact directly with any timecard that is currently displayed. Android comes with many built-in functionalities, such as sharing images, replying to message, all of which may be command through either the touch panel or said aloud. The GDK allows developers to create their own menu items, thus increasing customization and fluidity inside the software.

4.2 Mirror API[5]

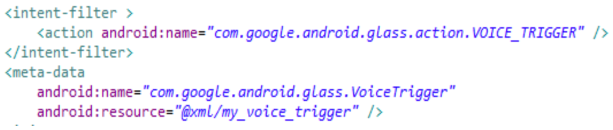
The Mirror API is a tool Google has created to allow developers to create services for Google Glass without having the software directly on the device. Google affectionately labels the modified software as “Glassware.” The Mirror API allows everything to function like a web based service, increasing productivity and versatility, it does all of this without actually running code on the Glass itself. The Mirror API serves as a means to simply display the data without too much having to cross through Google Glass' connection to the internet. It utilizes a REST API method which essentially means that instead of using arbitrary verbs to pull data it instead utilizes a simpler uniform selection of the HTTP methods 'GET', 'POST', 'PUT', and 'DELETE,’ but further detail into the precise command interfaces may be found on Google’s developers website. Since the project referenced in this paper was not created using the Mirror API, it would be interesting to note, the developer website also includes sample projects in the following languages: “Go[3],” “Java[2],” “.NET[6],” “PHP[7],” “Python[8],” and “Ruby[9].”

The functionality of the Mirror API differs only slightly for each aspect of the Glass interface. Since the Mirror API does not have hardware access its functionality is limited in regards to the Glass Development Kit. However, a Glass user will be able to push content to Glassware that is using the Mirror API method, regardless of hardware access. A Glass user simply has to tie their Google id to Glassware that uses the Mirror API. The Glassware will then start to push content to Google’s servers which will filter content to the appropriate end user. In addition, the Mirror API handles any displaying of data necessary in any glassware action. As was previously stated, the following project does not implement the Mirror API.

5. Project

The project encompasses the steps from creating a Glass application, “Glassware,” to the finalization of the publishing of the finished project. As well as outlining the essential pieces implemented for functionality. The goal of this project is to improve the everyday life of the hearing impaired. We developed this with the hopes of evolving into an all-around accessibility application. For now, the application caters to the hearing impaired. This project will act as the middleman between the speaker and the user. Much like movies have captions for the hearing impaired, our application will provide captions to the user for people around them.

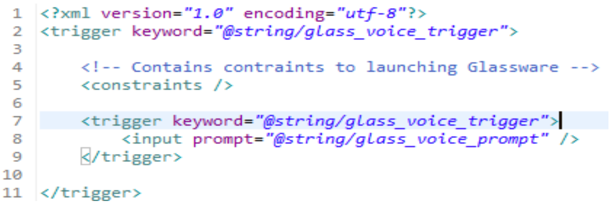
When assessing which development tools to use, we knew using the GDK would be the most reasonable. This will allow our application to be used and accessed by the user even if no phone is connected or no data connection is available by using the built in voice recognition software. Due to the constraints of the deployment environment. We wanted to design the application to run only when the user wanted it. We opted for an activity based approach over a service based approach. This requires the user to launch the activity every time they need it but saves battery and allows the user to control what actually gets displayed. The user will queue up the application via the Glass voice menu. Since every application within the voice menu must contain a voice trigger we wanted to make ours easy and simple to pronounce.



```
<intent-filter >
    <action android:name="com.google.android.glass.action.VOICE_TRIGGER" />
</intent-filter>
<meta-data
    android:name="com.google.android.glass.VoiceTrigger"
    android:resource="@xml/my_voice_trigger" />
```

Figure 3. XML queuing voice trigger.

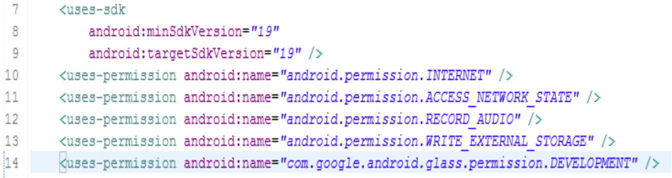
The voice trigger used by the Glass operating system is declared in the main AndroidManifest.xml file on a per activity base. The meta data contains the string information to be prompted to the user. The meta data tag points to our xml file located in our applications resource folder titled xml.



```
<?xml version="1.0" encoding="utf-8"?>
<trigger keyword="@string/glass_voice_trigger">

    <!-- Contains contraints to launching Glassware -->
    <constraints />

    <trigger keyword="@string/glass_voice_trigger">
        <input prompt="@string/glass_voice_prompt" />
    </trigger>

</trigger>
```

Figure 4. Voice trigger content

my_voice_trigger.xml contains the actual voice trigger content. Since we have no constraints determining if we can launch our application we simply point to our string resources that we give to the voice recognition activity. The Glass trigger keyword is aptly set as "caption" and the prompt simply tells the user that the Glass is listening and ready to receive voice input. As of the Glass operating system XE16+, when they moved to the Android operating system KitKat, applications must use one of the approved Glass command words listed at https://developers.google.com/glass/develop/gdk/reference/com/google/android/glass/app/VoiceTriggers.Command or add a permission named development in the applications manifest file (allows any command keyword to launch your app) to allow application launches on Glass.



```
    <uses-sdk
        android:minSdkVersion="19"
        android:targetSdkVersion="19" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.glass.permission.DEVELOPMENT" />
```

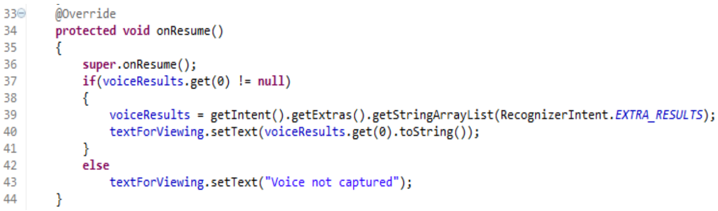
Firgure 5. Voice trigger development permission

5.1 Touch Interface

Glass also allows for touch interaction of the voice menu items. This allows the users to interact with our application without needing to use voice. The user will use our intuitive gesture system where they can take two fingers and swipe towards the Glass screen to queue up a new caption. The user will need to do this for each new caption they want to see. This allows for user control when they want it and for the application to conserve power when the user is not using it due to not having a background service running and not allowing the processor to go into deep sleep. The user interface will fit into the Glass design restrictions by being quick and easy to read.

5.2 Voice Recognition

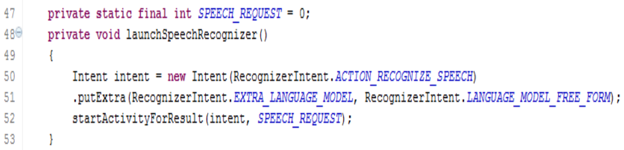
We call the voice recognition activity in the onResume() method of our application since the onCreate() method will call the onResume() method upon completion, this insures we gather voice data when the activity is resumed and created. We overwrite the main view screen each time the voice recognition activity is called if the results are not null.



```
@Override
protected void onResume()
{
    super.onResume();
    if(voiceResults.get(0) != null)
    {
        voiceResults = getIntent().getExtras().getStringArrayList(RecognizerIntent.EXTRA_RESULTS);
        textForViewing.setText(voiceResults.get(0).toString());
    }
    else
        textForViewing.setText("Voice not captured");
}
```

Figure 6. Calling Voice Recognition .

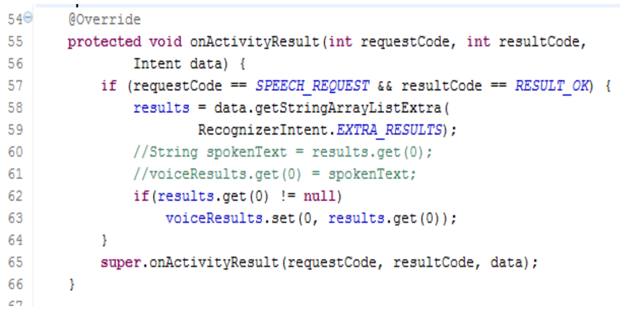
The user can queue up the Glass voice recognition activity manually from a menu item. This allows them to add captions when they want them and end a caption when they don't want them. We do this by passing an Intent to the voice recognition activity and waiting for a result from the VR activity.



```
private static final int SPEECH_REQUEST = 0;
private void launchSpeechRecognizer()
{
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
    .putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    startActivityForResult(intent, SPEECH_REQUEST);
}
```

Figure 7. Passing the Intent

launchSpeechRecognizer() is the method called by the menu listener that simply creates a voice recognition intent setting the language model to free form and starting the voice recognition activity for results.



```
@Override
protected void onActivityResult(int requestCode, int resultCode,
        Intent data) {
    if (requestCode == SPEECH_REQUEST && resultCode == RESULT_OK) {
        results = data.getStringArrayListExtra(
                RecognizerIntent.EXTRA_RESULTS);
        //String spokenText = results.get(0);
        //voiceResults.get(0) = spokenText;
        if(results.get(0) != null)
            voiceResults.set(0, results.get(0));
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

Figure 8. Getting results.

Upon getting the results from the activity it automatically will recall onResume() which will update the UI thread from the zero index of voiceResults string array, so upon results from the voice recognizer we set the zero index of the voiceResults to the new value received from the Glass VR. This is the application and how we handle the many interactions from the user.

6. Product release

As many may have recognized the stages are summarized as so: Stage 1, Stage 2, Profit. Luckily, this isn't so far off from the truth! The methods implemented by Google to publish and profit from your application is incredibly simple. All Glassware and, possibly, applications developed with GDK, will be distributed through Google's MyGlass website. As of this publication, Google has not yet released how they plan to receive GDK developed apps. However, if the Glassware is developed with the Mirror API, the Google develop website has a link to an online application form. Google is requiring Glassware to be submitted for review before they are published on the MyGlass website. Those who are developing with the Mirror API may wish to reference checklist that Google has posted online, that simply outlines the requirements and practices that are to be implemented before the review process.

Google has forbidden any commercial profit from Glassware and Glass applications, but upon reading the legal documentation on the MyGlass website, we expect this to change in either the near future or upon the general public release. The MyGlass has legal documentation prepared for Google Wallet, and if it is set up anything like the existing Google Play Market, then a developer will need to create a Google Wallet account and the funds will transfer directly into the account.

7. Conclusion

With the ease of programming and the predefined API interfaces, not to mention the sheer number of APIs provided, Google Glass market stands to be as massive and successful as the Android/Google Play is currently. The methods of programming for Glass, as noted above, serve as a breach into the new emerging mobile market, giving developers the means to potentially make a name for themselves, which could also prove to be quite lucrative. Google continues to innovate with newer pursuits in mobile computing, opening the world for the general user, allowing them access to tools that are virtually limitless. It is impossible to reject such innovations based on their own merit. Glass proves these statements true and leads the frontier in mobile computing, making the call for quality application that much more emphatic and necessary, truly, this is the future and for those reading, now is the time to break out from the wires of tradition and move into the new horizon of mobile computing.

8. References

[1] The Eclipse Foundation. (n.d.) Retrieved December 9, 2013 from the Eclipse website: http://www.eclipse.org/

[2] Oracle Inc. (n.d.) Retrieved December 9, 2013 from the Java website: http://www.java.com/en/

[3] The Go Authors. (n.d.) Retrieved December 9, 2013 from the Go website: http://golang.org/

[4] Google Inc. (n.d.) Retrieved December 9, 2013 from the Google Developer website: https://developers.google.com/

[5] Google Inc. (2013, November 22) Retrieved December 9, 2013 from the Glass Developer website: https://developers.google.com/glass/

[6] Microsoft Inc. (2008) Retrieved December 9, 2013 from the .NET website: http://www.microsoft.com/net

[7] ThePHPGroup. (n.d.) Retrieved December 9, 2013 from the PHP website: http://php.net/

[8] Python Software Foundation. (n.d.) Retrieved December 9, 2013 from the Python website: http://www.python.org/

[9] Ruby Community (n.d.) Retrieved December 9, 2013 from the Ruby website: https://www.ruby-lang.org/en/

Flipping the Classroom by Using Cloud Services and Third Party Online Courses in Traditional Class Settings Enables All Computer Science Students an Equal Education

D. Powell[1] **and J. Hollingsworth**[1]
[1]Department of Computing Sciences, Elon University, Elon, NC, USA

Abstract - *Massive Open Online Courses (MOOCs) and other online educational courses combined with free cloud platform as a service (PAAS) offerings provides a perfect environment for universities, large and small, public and private, to provide an equivalent state of the art undergraduate education to students. Traditional classroom classes can flip the classroom and have students review online content assignments outside of class and use class time for class discussion, assessment and team problem solving exercises. Elon University flipped the classroom for an Enterprise Programming Course during the spring 2014 semester using free and low-cost course online content material from codeschool.com and udemy.com with free cloud computing services from CloudBees. The flipped course was successful and will be tried at Elon in the fall with three different courses. This paper presents the advantages, disadvantages and challenges in using online resources and cloud computing as well as a discussion of lessons learned.*

Keywords: Cloud computing, Online courses, Flipping the classroom, Enterprise programming

1 Introduction

Graduating undergraduate computer science students need to understand mobile computing, desktop computing, enterprise computing and how to learn in a constantly changing world. Technology is rapidly changing and computer science faculty are highly challenged to offer the best and latest information in their undergraduate courses. University budgets are shrinking while costs of software, hardware and system administration are increasing. Despite the shrinking budgets and the rapid pace of change, the great news is that free cloud computing and free or low-cost online courses provides Computer Science faculty at all universities, public and private, large and small, the opportunity to offer their undergraduate majors a "Best in Class" education by flipping the classroom. The flipped classroom also known as flipped learning [1] inverts traditional teaching methods, delivering instruction online outside of class and moving class exercises and discussions into the classroom.

We believe that flipping the classroom by leveraging of online resources and the use of cloud computing is replicable across many of the standard computer science curriculum courses described in the Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science [2].

This hypothesis was proven at Elon University during the spring 2014 semester when we flipped the classroom and offered a class on enterprise programming. This paper describes the availability and use of online resources to provide content and the availability and use of cloud computing to provide the software and hardware for programming exercises and projects. This paper will present the advantages, disadvantages and instructor challenges in using online resources and cloud computing. The paper concludes with a discussion of lessons learned from flipping the classroom and future directions.

2 Online Courses

Most undergraduate computer science curriculums have similar courses based on the ACM Computer Science 2013 Curricula Guidelines, ABET 2013-2014 Criteria for Accrediting Computing Programs [3] and needs of their constituents. Typically, two of the expected program outcomes of a CS program are to provide students the skills for a lifetime of learning and to be successful in the field after graduation.

Many online resources are available from universities and from private companies that directly map or partially map to undergraduate core courses or elective courses. These online resources are available in the form of Massive Open Online Courses (MOOCs) from organizations and universities such as Coursera, Udacity, UC Berkeley and MIT [4] to high quality instructional videos from Udemy.com, Lynda.com, CodeSchool.com and Youtube.com. The challenge to the instructor is to find the online course or courses that have the materials that best match the content and instructional level for their class.

At Elon, a senior level, elective course on Enterprise Programming is offered every two years. The content of the course is focused on learning the latest technologies, best practices and best development techniques for designing and implementing a medium size enterprise application. Discussions with alumni indicated that the Java Spring Framework was the leading Java Framework in the enterprise field and would expose the Elon students to key concepts of Aspect Oriented Programming, Dependency Injection, Object Relational Mapping and Security. The department did not have any experience teaching the Java Spring Framework but wanted to offer the students the course. The authors searched for online courses and materials that they could quickly learn from experts in the field and in turn provide content for their traditional, face to face, twice a week class meetings for a fourteen-week semester. The authors found two online course providers and four courses from these providers to provide course content. The providers were codeschool.com and udemy.com. Two of the four courses, "Try JQuery" and "Discover DevTools", were from codeschool.com and two of the courses, "Servlets and JSPs: Creating Web Applications With Java" and "Java Spring 3.2 Framework" were from Udemy.com.

Selected online video course content was assigned as material to be completed before each class. Elon still had the traditional class, face to face 100 minute meetings twice a week. However, the class time was used for material highlighting, class discussion, and assessments instead of lecture. Elon found that the use of online video course content for a traditional course had many advantages, some disadvantages and some instructional challenges.

2.1 Online Course Advantages

The use of online course content has many advantages over traditional textbook driven classroom learning. Some of these advantages are the experience of the instructor, the high quality of the content, 24x7 availability, dashboard monitoring tools and low cost. An experienced, senior software developer and user of the technology on many successful and unsuccessful applications created the video content. The instructor can clearly identify best design and development practices. In Elon's case, the faculty member scheduled to teach the class had no previous experience with the Java Spring Framework. The video content developed by a software engineer with 14 years' experience allowed the Elon instructor to gain experience while providing expert content in a first time offered course to the students.

The selection of the video online content by the instructor was aided and enhanced via online reviews. The video online content had been vetted by previous students of the online course who had provided a rating of 1 to 5 stars along with comments on how well they liked the course. This rating system provided assurance of the quality and applicability of the course. In the case of the two Udemy courses used at Elon, the nine hour, Servlets and JSP course had a 5 star out of 5 star rating and had been taken by 6,300 students with 93 providing detailed review comments. The 27 hour, Java Spring Framework course had a 5 star out of 5 star rating and had been taken by 4,100 students with 92 supplying review comments.

The video online courses from major vendors are of high quality. They have been well designed, rehearsed and edited. The videos use voice, color, animation and live demonstrations to a degree that enhances learning and are in general superior to the typical university course taught by a faculty member using PowerPoint and live demonstrations.

The video online courses are available 24x7. They can be played from anyplace and at any time. Most of these courses have mobile applications so the course materials can be downloaded and played offline. From a learning perspective, students have the ability to stop, rewind and replay sections that they did not completely understand. The replay ability is also beneficial for students to catch up on missed classes due to an interview, illness or weather cancellations.

Most video online courses have a dashboard that monitors student progress. Faculty can monitor the individual student progress to identify students that are having issues or falling behind. Students can also interact with each other within or outside the university via online forums and wikis to ask questions, discuss programming issues or add additional insights. For example, codeschool.com has a dashboard that displays the course completion status of the logged in user for each course. The completion status shows the percentage of the course completed by the user. Completion consists of watching the video instruction and then successfully coding solutions to interactive programming challenges. Students were required to individually show at the start of designated face-to-face class meetings the completion status of "Try JQuery" and "Discover DevTools". The two Udemy courses also had a dashboard that showed the completion status of each lecture. The Servlets and JSP course had 7 main Sections made up of 60 video lectures and the Spring Framework had 16 main Sections made up of 169 video lectures. Typically, one section (consisting of two hours of video content) was assigned for completion, before each class session.

The cost of most online courses is very cheap compared to the purchase of textbook(s). In the case of codeschool.com, there is a monthly membership charge of approximately twenty dollars for access to all of their courses and some courses are completely free. The two courses used at Elon, "Try JQuery" and "Discover DevTools", were completely free. In the case of Udemy.com, the cost of the course is dependent on the instructor. At Elon, we contacted the instructor, and got a special rate of ten dollars a course.

2.2 Online Course Disadvantages

There were only two disadvantages to using online content. The first disadvantage is that students have traditionally learned from textbooks. The student is comfortable with coming to class for a lecture on a chapter, then reading the chapter, highlighting key points and completing end of chapter exercises. The online courses require more motivation and engagement on the part of the student to complete the online video "lectures" and to come to class prepared with questions. The superbly done video lectures are not repeated during class time but rather the class time is focused on meaningful discussion and assessment.

The second disadvantage is that a textbook is easier to reference than an online video course. Students need to learn to keep notes with appropriate reference hyperlinks during a video course to better review for a weekly quiz or monthly exam.

2.3 Instructor Challenges

The role of the instructor changes from one of primarily delivering lectures to one of organization and assessment. The key challenges are finding the right online course materials, finding and validating the supporting software, developing the syllabus and student motivation.

There are many online courses available and many coming out daily. The instructor is challenged to find and review the online course(s) materials that enhance the course content that have been vetted by the community. In some cases, there may be a single course that completely meets the course objective. However, in many cases, there may be multiple online courses that supply pieces that can be selectively chosen and woven into a better course.

The instructor needs to review or identify the course software packages, version numbers and code differences if any to be used to support the online course. Software changes rapidly and videos can quickly fall behind vendor releases of major new versions. The software used in class must match as close as possible the software used in the online content.

With respect to the Enterprise programming course at Elon, the key challenges to finding online content fell into deciding on a consistent Interactive Development Environment (IDE), test application server, MySQL development interface and client browser for all online content. Most online courses either use an IDE of Eclipse or NetBeans. The test application servers are usually Tomcat or Glassfish. The MySQL development environment is either phpMyAdmin or MySql Workbench. The browser and debugging tool is either Firefox with Firebug or Chrome with DevTools. Elon chose online content that matched the Udemy.com selected videos. The selected software was the Eclipse IDE, the Tomcat application server, MySQL Workbench and the Chrome browser with DevTools.

The instructor needs to develop the course syllabi for each semester class with due dates for viewing and completing selective online course content. The instructor needs to identify additional course material to supplement online course content. The instructor needs to develop assessment materials that include quizzes, class exercises, large projects and exams. For the Elon course, the seven quizzes, midterm exam and final exam were developed from industry sample certification exams for Oracle Java EE 6 Web Component Developer Certified Exam and SpringSource Web Application Developer. The certification tests were selected to provide external validation for the course coverage and learning of our students. The large number of video hours to complete the Udemy course required a decision to be made on whether to complete all lectures and get a Udemy "Certificate of Completion" or to stop at an earlier point to allow time for a semester project. Ideally both could have been done but, due to time constraints, a choice was made to complete all lectures and receive the completion certificate.

Finally and most importantly, the instructor needs to provide the motivation for each class to keep the student on track. The student is exposed to a different style of learning compared to in-class lectures. The class session is used to continually assess the student progress and understanding of the material and to provide team development exercises. To keep the student engaged, the student was required to demonstrate the working completion and deployment to the cloud of the software covered in the previous class assigned video materials. For the video materials assigned for the current class, the student had to show the course dashboard completion status.

3 Cloud Computing

Cloud computing offers Platform As A Service (PAAS) functionality for software deployment. Over the past three years, Elon has successfully used free cloud computing from Google App Engine, IBM Smart Cloud Enterprise and CloudBees to support classes on web development, mobile android device development and enterprise programming [5][6][7]. The most recent course on Enterprise programming used CloudBees as the cloud provider. Cloud computing offers many advantages and few disadvantages.

3.1 Cloud Computing Advantages

Cloud computing offers all universities a chance to provide free access and deployment using the latest software and hardware. Universities are no longer constrained based on budget or in house administrative support. The advantages are:

1. There is no cost for the hardware and software. Each student can register for a free account that comes with a free quota of file storage, database storage, application programming interface calls (API) and

emails sent. The quotas are generous and have never proven to be a constraint.

2. The hardware and software is available 24x7. It is administered by the cloud vendor with frequent updates and patches. The cloud vendor provides extensive documentation and examples of how to use the cloud. Each cloud vendor is used by thousands of developers. As such, the cloud is very reliable and dependable for course instruction. In the past three years, the Elon student has not lost any time due to the Cloud vendor not being available for deployment.

3. There are no internal firewall issues for code deployment. The student can deploy code from school, from home or from anywhere with an internet connection. The student can easily demonstrate developed web applications or web services to employers.

4. With Google App Engine and CloudBees each student signs up for their own account. Each student account has its own security and sandbox. The source code development and work of one student is not visible or accessible to another. This helps to prevent the unauthorized sharing of code among students. The account will stay with the user after the course and after graduation. This allows students to create a portfolio of projects from different courses to show employers during interviews. Unfortunately, the IBM Smart Cloud Enterprise had individual student accounts that only lived for the duration of the course. This proved to be a problem for students that wanted to demonstrate their projects at job interviews after the semester completed.

5. Each of the three listed cloud vendors provides support for Java Enterprise standards for servlets, java server pages and access to a back end data store. However, if MySQL is used as the back end database on Google App Engine then there is a minimal cost for storage.

3.2 Cloud Computing Disadvantages

Each of three PAAS offerings has proven very reliable for students and faculty. However, there is one large disadvantage of using a commercial offering and one small disadvantage.

The large disadvantage is the dependence of the course success and student success on the cloud platform being freely available. If the company stops offering the cloud service or changes the costing structure and eliminates free usage in a middle of a semester then this would be disastrous to the course. On the positive side, the Google App Engine cloud has been around for over three years and the free cost structure has not changed. The IBM Smart Cloud was available for two years and then IBM announced they would no longer support it as of Feb 2014 and would replace it with a new offering called BlueMix. This new offering is currently in Beta and will be available in 2015. IBM did notify all impacted universities four months in advance so no in session fall courses were directly impacted. However, faculty teaching in the spring had to find a new cloud provider. In the case of Elon, we chose CloudBees and have been very impressed. It is our planned platform of choice for fall 2014 offerings. CloudBees fully supports the Java Enterprise 7 standards and provides free access to Tomcat, Glassfish, JBoss and Wildfly application servers. In addition, it provides free access to a MySQL database and free access to a mail server for sending email.

The one small disadvantage is a vendor specific implementation often has limitations on the Java Enterprise Standards implementation. We ran into this limitation with respect to the Google App Engine which supports most of but not all of Java Enterprise Edition 7. Servlets and Java Server Pages are directly supported but Java Server Faces is not. Other APIs that are not fully supported are File IO and JPA.

3.3 Instructor Challenges

The main challenges in selecting a cloud service provider is to identify one that is widely used, well documented, free, supports Java Enterprise 7 and provides access to JavaMail and an external SQL database. Once a provider is selected then there is a significant time commitment invested by the faculty member to understand the vendor specific code deployment configurations required by the vendor.

4 Conclusions

The large number of free or low cost online course content and free use of cloud PAAS computing running industry standard Java Enterprise Edition software provides every university access to the very best instruction and computing. The challenge to faculty is to identify the best content and to flip the classroom to have students watching and interacting with online course content outside of class. Instead of delivering lectures, faculty will spend class time focusing on question and answer sessions, group exercises and assessment. Faculty have dashboard and monitoring tools to best identify the students that need more attention and provide them additional feedback.

At the end of the Elon spring semester, the twelve students in Enterprise Programming were anonymously surveyed for their thoughts on our experiment with flipping the classroom. Nine of the twelve students (75%) enjoyed the format of flipping the course from the standard class lecture, two of the students (16.6%) were not sure if they preferred it and one student (8.3%) preferred the traditional lecture format. However, seven of the twelve students (58%) indicated they would have preferred more in class problem solving exercises and an end of course project to pull all the concepts together. Students suggested that the in class exercises could be extensions to the work done in the previewed online videos or a completely different problem solving exercise to reinforce the topics covered in the videos. The authors completely agree with the student suggestions that more problem solving class exercises and a culminating project would have been beneficial. The authors made a tradeoff to cover more material at the expense of fewer exercises and in hindsight should have compromised on the amount of material covered and included more exercises.

Based on the lessons learned in flipping this prototype class, three core classes in the fall 2014 will be partially flipped with a percentage of the classes in each course flipped and a percentage kept in the traditional format. The percentages will be based on an analysis of the online materials available and how well they match the course objectives. These core classes will be for Computer Science II, Computer Science III and the Computer Science Capstone. Second year, Computer Science II students learn object oriented java programming on a desktop. Third year, Computer Science III students learn basic web programming with HTML, CSS, javascript, servlets, java server faces and connecting to a back end, SQL database. Fourth year, Computer Science Capstone students learn design patterns for mobile, desktop and enterprise applications.

In closing, the authors believe the use of the latest professionally developed online content combined with cloud computing and in class exercises and assessment will allow Elon to offer courses that are as good as any university with any budget anywhere in the world. This practice can be replicated at any other large or small university.

5 References

[1] N. Hamdan, P. McKnight, K. McKnight, and K. Arfstrom, "The Flipped Learning Model: A White Paper Based On the Literature Review Titled A Review Of Flipped Learning", 2013. Available: http://researchnetwork.pearson.com/wp-content/uploads/WhitePaper_FlippedLearning.pdf

[2] ACM, Computer Science Curricula 2013, Available: http://www.acm.org/education/CS2013-final-report.pdf

[3] ABET, Criteria for Accrediting Programs, 2013-2014, Available: www.abet.org/DisplayTemplates/DocsHandbook.aspx?id=3148

[4] MOOCs: Top 10 Sites for Free Education With Elite Universities. Available: http://www.bdpa-detroit.org/portal/index.php/comittees/high-school-computer-competition-hscc/29-education/57-moocs-top-10-sites-for-free-education-with-elite-universities.html.

[5] J. Hollingsworth, and D. Powell, "Seeding the Cloud with Students from Every Computer Science Undergraduate Program", ICEIS 2013 – Proceedings of the 15th International Conference on Enterprise Information Systems, Volume 2, SciTePress, 2013, pp 513-518.

[6] D. Powell and J. Hollingsworth, "Google and IBM Cloud Make Enterprise Computing Available to all Undergraduate CS Majors", Proceedings of the 2012 International Conference on Frontiers in Education: Computer Science and Computer Education. 2012, pp 611-616.

[7] J. Hollingsworth and D. Powell, "Teaching Web Programming Using the Google Cloud," Proceedings of the 48th ACM Southeast Conference, 2010, pp 76-80.

Preparing Digital Natives for Industry

A. Dr. Neelu Sinha[1], **B. Mary Rauscher**[2]
[1]Department of Mathematics, Computer Science and Physics, Fairleigh Dickinson University, Madison, NJ, United States
[2]Graduate Student, Fairleigh Dickinson University, Madison, NJ, United States

Abstract - *A huge skills gap crisis is raging and employers are unable to find graduates with necessary workplace skills for 21st century. More than 75% of employers want colleges to place more emphasis on skills that help students develop five key learning outcomes: critical thinking, complex problem-solving, written and oral communication, and applied knowledge in real-world settings. Discussions with college alumni further shed light on the significance and urgent need for such an education. This research explores ways to increase the versatility of college graduates to gain a strong foothold in the job market by enabling them to gain these skills while in college. A student embarked upon a project to become an active creator of technology rather than just being passive consumer of technology by developing applications ("Apps") for handheld mobile devices on various platforms. Such a multi-platform approach facilitates our graduates to garner vital skills required by employers.*

Keywords: Teaching Digital Natives, Mobile Learning and Use of Mobile Technologies, Consumers of Technology to Creators of Technology, Experiential Education, 21st Century Skills

1 Introduction

We are facing a crisis in the United States whereby we have a huge skills gap and employers are unable to find graduates with necessary workplace skills for the 21st century. More than 75% of employers say they want colleges to place more emphasis on workplace skills that help students develop five key learning outcomes, including: critical thinking, complex problem-solving, written and oral communication, and applied knowledge in real-world settings. Fischer (2013) points out that more than half the employers had trouble finding recent graduates qualified to fill positions, as the graduates lacked basic workplace proficiencies: adaptability, communication skills, and the ability to solve complex problems. This implies that we are not adequately preparing our graduates in written and oral communication, decision-making, and analytical & research skills. Employers also place a high priority on forms of learning that emphasize a students' active development and application of intellectual and practical skills to real world problems (Hart Research Associates, 2013).

1.1 Choice of a College Major

Carnevale and Cheah (2013) argue that not all college majors are created equal. Choice of major significantly affects employment prospects and earnings. Some majors offer substantially better employment prospects than others. In particular, people who make technology (creators) are better off than people who simply use technology (passive consumers). In general, majors that are linked to occupations have better employment prospects than majors focused on general skills.

1.2 Skills Gap

Employers claim that they are unable to find enough skilled workers in the United States implying a skills shortage. The OECD (OECD, 2013) report assesses literacy, mathematical, and problem-solving skills in people aged 16 to 65 in 22 advanced nations of the OECD and Russia and Cyprus. Based on this OECD report, Perez-Pana (2013) cautions that American adults lag well behind their counterparts in most other developed countries in the mathematical and technical skills – these skills are so essential in a modern workplace. He also cites a statement from Arne Duncan, the education secretary, saying that the findings "show our education system hasn't done enough to help Americans compete - or position our country to lead - in a global economy that demands increasingly higher skills." Porter (2013), also based on the OECD report, disagrees with Washington and the United States Chamber of Commerce that American workers are the best in the world and claims the skill level of our American labor force has fallen dangerously behind its peers around the world. So, what are these skills? How can we help our graduates attain these skills and be ready for a job in the industry?

1.3 21st Century Skills

The need for routine cognitive and craft skills in the services sector is diminishing rapidly as many of the manufacturing and low-skill tasks are being automated due to the advent of Information and Communications Technology (ICT). In addition to the "occupation-specific" skills, employers today are also demanding:

- "generic" skills such as information-processing skills (including literacy, numeracy and problem solving skills) and

- “soft” skills such as interpersonal communication, self-management, and the ability to learn new skills.

A Framework for 21st Century Learning, developed by the Partnership for 21st Century Skills (2011) P21 describes skills, knowledge and expertise students must master to succeed in work and life, which includes not only the 3Rs (the core subjects) and the 4Cs (Critical Thinking, Communications, Collaboration and Creativity) but also the Information, Media & Technology skills (the “generic” skills) and Life and Career skills (“the “soft” skills) (see Figure A1). What can we educators do to enable our students to master these 21st century skills using tools currently at our disposal? Is there a way to enrich our students with generic and soft skills with contemporary devices based on various platforms?

2 Mobile Apps Development

An App (or Application), for a computer or phone operating system, is a self-contained program or piece of software designed to fulfill a particular purpose. Apps are increasingly popular for hand-held mobile devices (smartphones and tablets) which are available for different platforms ranging from Google’s Android, Apple’s iOS to Microsoft’s Windows operating systems. In today’s global marketplace employers offer products and services for all such devices (on different platforms) and thus are demanding that our graduates be versatile and well-versed with all the available platforms (a 21st century skill), not just as consumers but as creators of technology.

2.1 Mobile Learning

Sales of hand-held mobile devices are exploding and drastically altering our education landscape by revolutionizing the ways we teach and learn using these on-line, social, and increasingly mobile computing devices which are ubiquitous and offer visual, tactile and personal experiences as never before. Figure A2 (Sanou, 2013) illustrates the growth in mobile phone subscribers worldwide from under a billion in 2000 to about 6 billion in 2011. This number is close to 7 billion in 2012, which is about 96% of our global population! For the first time in history, a majority of people can afford to buy personal ICT in the form of mobile devices, in particular mobile phones (Vosloo, 2012). These devices are digital, portable and provide multimedia capabilities to access the Internet. Our students today are “Digital Natives” – a term coined by Prensky (2009) –“native speakers” of the digital language of: computers, video games and the Internet. To speak the language of these “natives” we use an innovative approach - mobile application development for different platforms, so that they can experience mobile technology as creators of technology, and not just as consumers.

2.2 Mobile Technology

The traditional classrooms (chalk and blackboard) of yesterday are shrinking into the palms of our hands today! E-Learning including Computer/Web based training, technology enhanced learning, multimedia (Audio Video) learning, online classes, distance learning, virtual classes, etc. is readily available today due to advances in electronic media and ICT. Mobile technology impacts learning that takes place outside the “traditional classroom” and has the power to transform education. Mobile Learning represents a profound shift in the way education is delivered and received. It is not just distance learning with mobile technology but a union of ICT and education where technology-enabled learning solutions are available to learners anytime, anywhere. UNESCO (Vosloo, 2012) believes that this unification has great potential to facilitate knowledge dissemination, improve learning and assist the development of more efficient education services. Any portable device, such as a tablet, laptop or mobile phone that provides access to educational content through mobile connectivity can be a tool for Mobile Learning. As prices of mobile devices and connectivity continue to plummet, Mobile Learning can extend educational opportunities to marginalized groups; increase education quality; and reduce inequalities based on gender, class, race, age and disabilities (Vosloo, 2012).

2.3 Challenges

Aside from the immense benefits, Mobile Learning is burdened with several challenges. Negative perceptions around use of mobile devices (especially phones) in educational settings; safety, security and privacy concerns; and lack of training opportunities for educators to incorporate mobile learning in their classrooms further thwart innovation. Also, development needs as well as the hardware and equipment needs vary from platform to platform, further widening the skills gap.

To address these challenges and to overcome the skills gap we primed our student to design and develop mobile apps for the various platforms such as: Apple, Inc’s iOS, Microsoft’s Windows Phone Development and Google’s Android operating systems. For this our student had to learn three different programming languages and understand the different platforms they were based on. This enabled our student to adjust to any platform (a 21st century skill desired by employers) and be ready for the workplace.

2.4 Apple iOS App Development

iOS is the foundation of iPhone, iPad, and iPod touch and apps for these devices are developed using the Objective-C language using Apple’s Integrated Development Environment (IDE) Xcode (Conway, Keur and Hillegass 2014 and Kochan 2013). Xcode comes with an emulator which can simulate an iPhone as well as iPad and is used to test the code for an app under development.

2.5 Microsoft Windows 8 Phone App Development

Windows 8 apps are written in the C# programming language (Sharp, 2013). The Windows Phone 8 Software Development Kit (SDK) is needed to create Windows Phone 8 Apps, and it comes with Microsoft's Visual Studio Ultimate 2013. This is an excellent IDE with many options for creating and customizing User Interface (UI) items for the app.

2.6 Google Android App Development

Android apps are written in the Java programming language and use the Eclipse IDE Eclipse (Harwani 2013). Google also has an excellent online Android Application Programming Interface (API) which serves as a reference for learning this new platform.

3 Results and Analysis

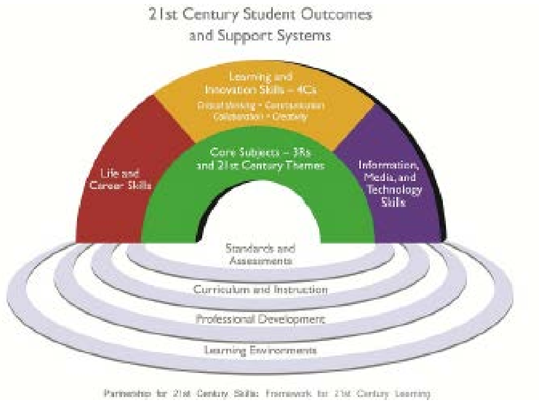
Each platform has its own development framework and tools which the student has to learn and master before attempting to develop real-world apps. The learning curve for each of the platforms varies depending on the student's background with that environment. Learning a programming language can be both quite interesting and challenging. Most colleges have requirements of at least three different programming languages. If a student is already well-versed with a particular programming language then there is a definite advantage to using that platform and thus having a short learning curve.

For students new to a programming language some lead time is spent learning the syntax and semantics of the new language. This could lead to frustration on part of the student since creation would not happen for a long time until they have consumed a lot of information about the language peculiarities. A lot of time is spent in writing sample programs simply to understand the language basics. Many times it helps if a student can think in simple English (or English-like) language and trace out a detailed path using an algorithm or pseudo code to guide the process of generating code. Flow charts and decision trees also help in the process of coming up with new ways to solve a problem. A study of some common programming paradigms, especially procedural and object-oriented, also serves to be very helpful. In order to formulate real-world apps, the student needs to understand the data structure and learn key subjects like searching, sorting, binary tree and linked list creation. Further, computer algorithms like divide and conquer, greedy method, backtracking, etc. should also be a part of the learning process. Student needs to understand that even the most accomplished programmers may fail to solve simple problems at times. So, finding or preparing a proper algorithm, learning data structures, and many hours of practicing problem-solving skills on different types of problems is essential.

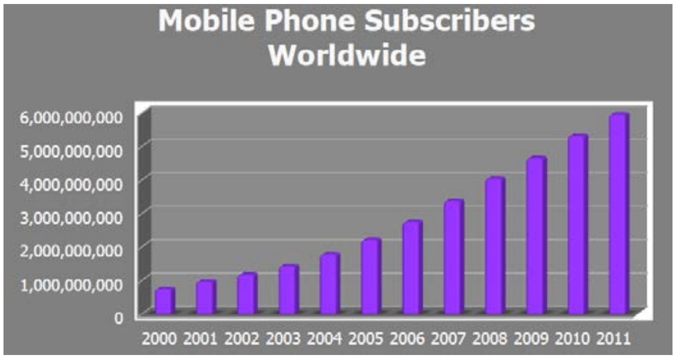
Computational skills are exercised while programming and it can be very frustrating thus having a solid foundation in mathematics and logic helps with the process of learning to program to solve a problem. Various mathematical topics – Equations, Calculus, Fourier, Vector, Boundary Value Problem, Linear Algebra, Matrices, etc. will come in handy while programming.

A student benefits tremendously from teaching others and sharing developed apps with others on the different platforms. However, a major impediment is the cost associated with the different platforms – these include the costs of the hardware (computers and mobile devices), developer's fees charged by the different platform authorities, books, training manuals, and other documentation. A way to mitigate some of these costs involves forming a partnership with potential employers and industries in the form of internships. The employers may already have the necessary hardware for the different platforms thus reducing the cost to the educator. Such experiential learning is beneficial to both the learner as well as the employer. Also, internships would put our student at the forefront of technology and we educators would be better able to understand the ever-changing needs of the industry.

3.1 Figure A1 - 21st Century Student Outcomes and Support Systems


21st Century Student Outcomes
and Support Systems
Learning and Innovation Skills – 4Cs
Critical thinking • Communication
Collaboration • Creativity
Life and Career Skills
Core Subjects – 3Rs and 21st Century Themes
Information, Media, and Technology Skills
Standards and Assessments
Curriculum and Instruction
Professional Development
Learning Environments
Partnership for 21st Century Skills: Framework for 21st Century Learning

3.2 Figure A2 - Growth in Mobile Phone Subscribers – Worldwide



4 Conclusions and Future Work

This research explores ways to increase the versatility of graduates so that they can gain a strong foothold in the job market and become active creators of technology rather than just being passive consumers of technology. The employers today are indeed demanding students with a well-rounded education who are “digitally ready” to jump into their workplace responsibilities. Using mobile technologies in this Mobile Apps Development study we have been able to help our graduates evolve from “consumers” to “creators” of technology and gain computational thinking skills at the same time. Interactions with various assessment tools and maintaining electronic portfolios also gives our students an edge so that they can be comfortably “digitally ready” to enter into the workplace. For the future we wish to incorporate experiential learning into our course so students can team up with local businesses/employers to gain more hands-on training while trying to design an app for these folks based on the specific industry they are involved with. A multi-platform approach facilitates our graduates to garner vital skills required by employers thereby diminishing the skills gap in a global economy that demands increasingly higher skills.

5 References

[1] Carnevale, A. P. & Cheah, B. (2013). Hard Times: College Majors, Unemployment and Earnings: It Still Pays to Earn a College Degree But Not All College Degrees are Created Equal. *Georgetown University Georgetown Public Policy Institute Center on Education and the Workforce*. Retrieved from http://www9.georgetown.edu/grad/gppi/hpi/cew/pdfs/HardTimes.2013.2.pdf

[2] Conway, J., Keur, C., and Hillegass, A. (2014). *iOS Programming: The Big Nerd Ranch Guide*. Atlanta, GA: Big Nerd Ranch. 2014.

[3] Fischer, K. (2013). *The Employment Mismatch: A College Degree Sorts Job Applicants, but Employers Wish It Meant More*. The Chronicle of Higher Education: Special Reports. Retrieved from: http://chronicle.com/article/The-Employment-Mismatch/137625/#id=overview

[4] Hart Research Associates (2013). *It takes more than a major: Employer priorities for college learning and student success*. An online survey among employers conducted on behalf of: The Association Of American Colleges and Universities. Washington, DC. Retrieved from http://www.aacu.org/leap/documents/2013_EmployerSurvey.pdf.

[5] Harwani, B. M. (2013). *Android Programming Unleashed*. Indianapolis, IN: Sams, 2013.

[6] Kochan, S. G. (2013). *Programming in Objective-C*. Upper Saddle River, NJ: Addison-Wesley, 2013.

[7] OECD (2013). *Organization for Economic Co-operation and Development (OECD) Skills Outlook 2013*: First Results from the Survey of Adult Skills, OECD Publishing. doi: 10.1787/9789264204256-en

[8] Partnership for 21st Century Skills (2011). *Framework for 21st Century Learning*. Retrieved from: http://www.p21.org/storage/documents/1.__p21_framework_2-pager.pdf

[9] Perez-Pana, R. (2013). *U.S. Adults Fare Poorly in a Study of Skills*. The New York Times. Retrieved from: http://www.nytimes.com/2013/10/08/us/us-adults-fare-poorly-in-a-study-of-skills.html?_r=1&

[10] Porter, E. (2013). *Stubborn Skills Gap in America’s Workforce*. The New York Times. Retrieved from: http://www.nytimes.com/2013/10/09/business/economy/stubborn-skills-gap-in-americas-work-force.html

[11] Prensky, M. (2009). H. sapiens digital: *From digital immigrants and digital natives to digital wisdom*. Innovate 5 (3). Retrieved from: http://www.innovateonline.info/index.php?view=article&id=705

[12] Sanou, B. (2013) The world in 2013 – ICT Facts and Figures. *ICT Data and Statistics Division, Telecommunication Development Bureau, International Telecommunication Union*. Retrieved from: http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx

[13] Sharp, J. (2013). *Microsoft Visual C# 2013 Step by Step*. Microsoft Press; 1 edition, 2013.

[14] Vosloo, S. (2012). Mobile learning and policies – key issues to consider. The United Nations Educational, Scientific and Cultural Organization UNESCO’s Working Paper Series

on Mobile Learning. ISSN 2227-5029. Retrieved from: http://unesdoc.unesco.org/images/0021/002176/217638E.pdf/

Health Journal Web Service using Cloud Computing

Ryan Hausen, **Samuel Sambasivam**, **Simon H Lin**
Department of Computer Science, College of Liberal Arts and Sciences,
Azusa Pacific University, Azusa, CA 91702, USA

Abstract - *Technology continues on a trend to a stronger dependence on the cloud. Making web services easy to create and easy to manage has taken large strides in recent years, and there has been a push for more RESTful web services [3]. In this same stride, software has been leaning towards the implementation of software as a service which can utilize RESTful web services to access databases or other backend services. In this paper, we explore how to implement a simple RESTful web service using the recently released Microsoft Web API 2.1 backed by SQL server. Using this impressively easy technology with features like attribute routing and the new IHttpActionResult interface, receiving and sending HTTP requests and responses comes very naturally to anyone who is familiar with .NET programming and some standard web protocols.*

Keywords: web service, software design, programming, database, cloud computing

1 Requirement

The outcome of this project will be a RESTful web service, henceforth referred to as *the Service*. The Service will support a Health Journal storage system, by which users can submit a daily journal and retrieve it later. Each journal should be able to track the following six pieces of information:

1. Date that the journal represents
2. How the person was feeling that day
3. How much the person weighs that day
4. How far the person traveled in exercise that day
5. How many calories that person consumed that day
6. Any extra notes that person would like to associate with that day

In addition to being able to submit data to the Service, the Service should be able to respond with a single journal or a series of journals filtered by date so that users can monitor their results over time.

The Service will provide the basic Create, Read, Update, and Delete functions. These will be exposed in intuitive functions that will make it easy for client software to utilize its features for the benefit of users.

2 High-Level Design

The design pattern that the Service architecture should follow will be based on REST or Resource Oriented Architecture. In this design pattern, the resources are represented by URI's that utilize standard HTTP methods (POST, PUT, GET, and DELETE) to perform operations against the data represented by the URI, at which the methods are applied.

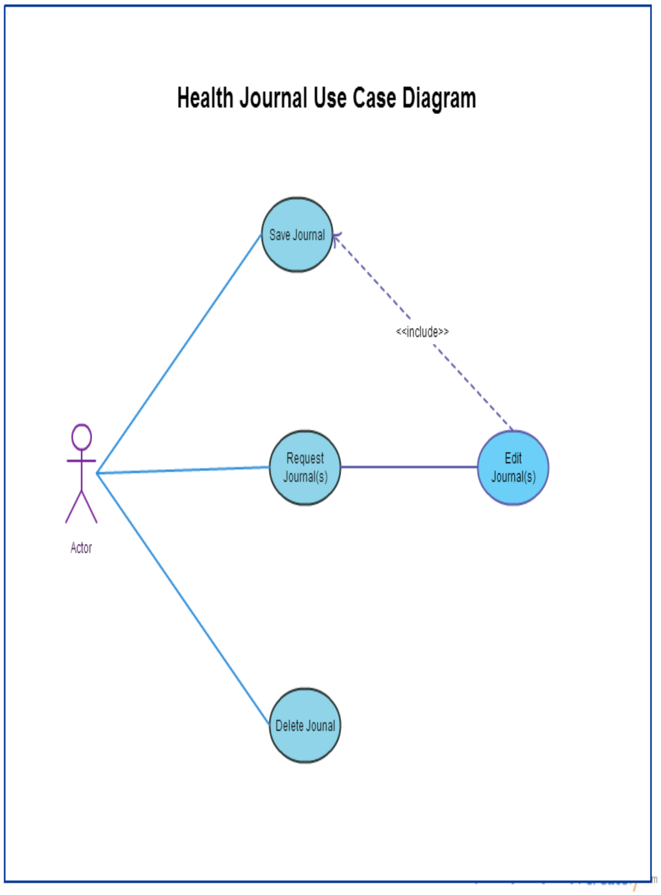


Figure 1: Use Case Diagram for Managing Health Journals

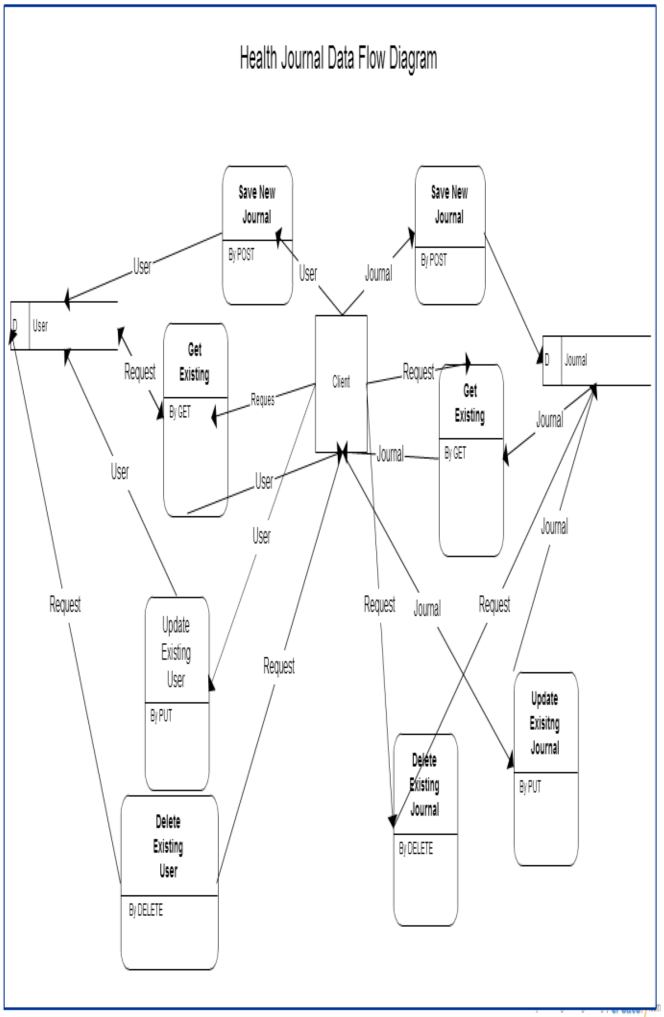


Figure 2: Data Flow Diagram for the Health Journal Web Service

3 Low-Level Design

The particular implementation of the RESTful architecture pattern for this project will be Microsoft's Web API 2.1 in the .Net environment, and using C#/SQL as the languages of implementation. The target framework [1] for the project is .Net 4.5 and will be hosted on a Microsoft IIS server. The data will be stored using Microsoft's SQL Server 2012.

The dominant format of data on the web currently is XML. Another format, JSON is growing in popularity. JSON is a lightweight form of XML that simplifies data transmission in a way that is easily understood in JavaScript and consequently most browsers.

The class diagram has been designed with all attributes [4] and relationships verified and validated. The data model has been designed with all attributes and relationships verified and validated [5]. The class diagram and the database diagram are shown below.

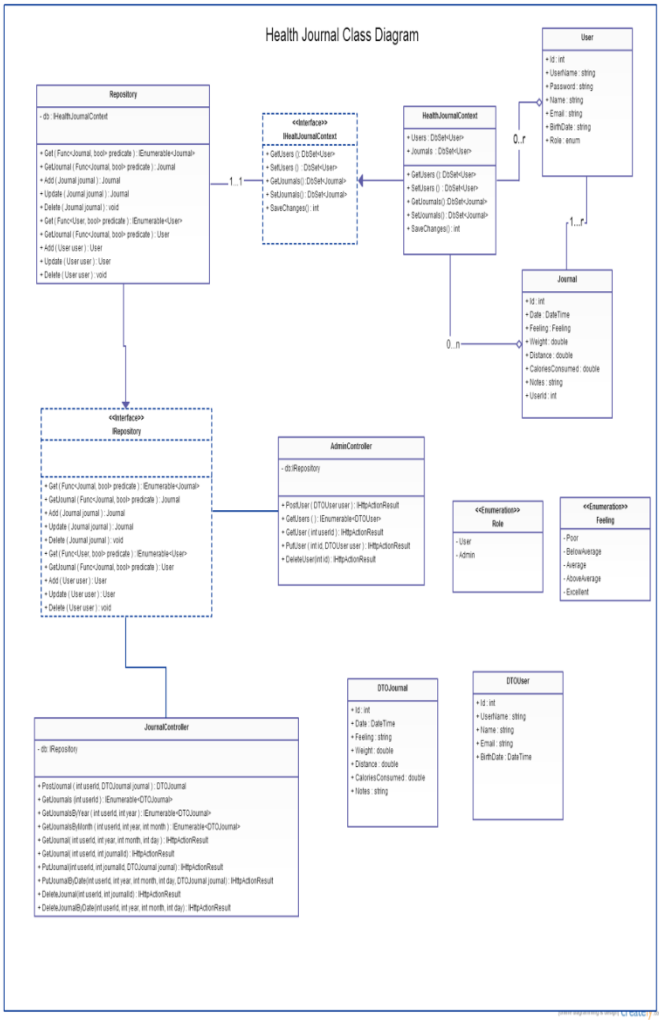

Health Journal Class Diagram
Repository
- db : IHealthJournalContext
+ Get (Func<Journal, bool> predicate) : IEnumerable<Journal>
+ GetJournal (Func<Journal, bool> predicate) : Journal
+ Add (Journal journal) : Journal
+ Update (Journal journal) : Journal
+ Delete (Journal journal) : void
+ Get (Func<User, bool> predicate) : IEnumerable<User>
+ GetJournal (Func<Journal, bool> predicate) : User
+ Add (User user) : User
+ Update (User user) : User
+ Delete (User user) : void
1...1
<<Interface>>
IHealtJournalContext
+ GetUsers (): DbSet<User>
+ SetUsers () : DbSet<User>
+ GetJournals() :DbSet<Journal>
+ SetJournals() : DbSet<Journal>
+ SaveChanges() : int
HealthJournalContext
+ Users : DbSet<User>
+ Journals : DbSet<User>
+ GetUsers (): DbSet<User>
+ SetUsers () : DbSet<User>
+ GetJournals() :DbSet<Journal>
+ SetJournals() : DbSet<Journal>
+ SaveChanges() : int
0..r
User
+ Id : int
+ UserName : string
+ Password : string
+ Name : string
+ Email : string
+ BirthDate : string
+ Role : enum
1...r
0...n
Journal
+ Id : int
+ Date : DateTime
+ Feeling : Feeling
+ Weight : double
+ Distance : double
+ CaloriesConsumed : double
+ Notes : string
+ UserId : int
<<Interface>>
IRepository
+ Get (Func<Journal, bool> predicate) : IEnumerable<Journal>
+ GetJournal (Func<Journal, bool> predicate) : Journal
+ Add (Journal journal) : Journal
+ Update (Journal journal) : Journal
+ Delete (Journal journal) : void
+ Get (Func<User, bool> predicate) : IEnumerable<User>
+ GetJournal (Func<Journal, bool> predicate) : User
+ Add (User user) : User
+ Update (User user) : User
+ Delete (User user) : void
AdminController
- db:IRepository
+ PostUser (DTOUser user) : IHttpActionResult
+ GetUsers ( ) : IEnumerable<DTOUser>
+ GetUser (int userId) : IHttpActionResult
+ PutUser (int id, DTOUser user) : IHttpActionResult
+ DeleteUser(int id) : IHttpActionResult
<<Enumeration>>
Role
- User
- Admin
<<Enumeration>>
Feeling
- Poor
- BelowAverage
- Average
- AboveAverage
- Excellent
JournalController
- db: IRepository
+ PostJournal (int userId, DTOJournal journal) : DTOJournal
+ GetJournals (int userId) : IEnumerable<DTOJournal>
+ GetJournalsByYear (int userId, int year) : IEnumerable<DTOJournal>
+ GetJournalsByMonth (int userId, int year, int month) : IEnumerable<DTOJournal>
+ GetJournal(int userId, int year, int month, int day) : IHttpActionResult
+ GetJournal(int userId, int journalId) : IHttpActionResult
+ PutJournal(int userId, int journalId, DTOJournal journal) : IHttpActionResult
+ PutJournalByDate(int userId, int year, int month, int day, DTOJournal journal) : IHttpActionResult
+ DeleteJournal(int userId, int journalId) : IHttpActionResult
+ DeleteJournalByDate(int userId, int year, int month, int day) : IHttpActionResult
DTOJournal
+ Id : int
+ Date : DateTime
+ Feeling : string
+ Weight : double
+ Distance : double
+ CaloriesConsumed : double
+ Notes : string
DTOUser
+ Id : int
+ UserName : string
+ Name : string
+ Email : string
+ BirthDate : DateTime

Figure 3: Class diagram for the Health Journal Web Service

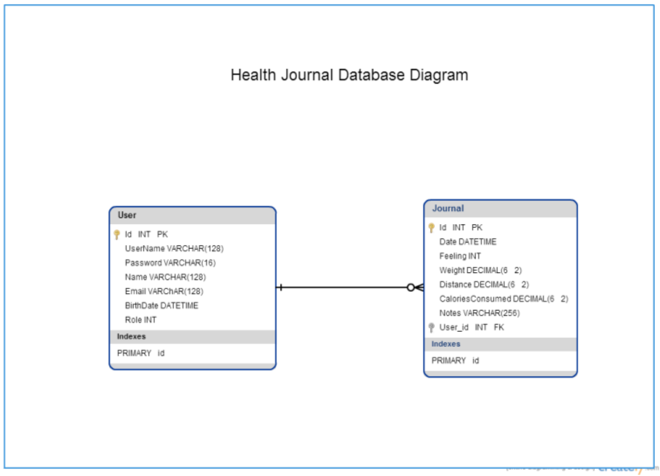


Figure 4: Database Diagram for the Health Journal Web Service

4 Coding

The coding has been completed successfully with approximately 3000 lines of code. The complete source code is available upon request.

5 Testing

Unit testing [2] and system testing have been completed accordingly with some complex testing scripts being designed and developed.

6 Product Demo

To demonstrate the Service, Dev HTTP Client, available in the Google Chrome App Store was used to generate HTTP requests and parse responses. To find out more about Dev HTTP client, please go to: https://plus.google.com/104025798250320128549.

The following is a sample list of actions a user or a service administrator can perform.

1. *Create a new user:* A new user is created using JSON format and sent using the POST method. Note that the server response excludes the new user's password and includes the user's Id. The response also includes the URI for new user in Location Header.
2. *Add a new journal to the created user's journals:* A new journal is created using JSON format and sent using POST, referencing the created user's Id in the URI.
3. *Retrieve an existing journal:* An existing journal is retrieved by making a GET request to the URI of the journal.
4. *Update an existing journal with new information:* An updated journal is sent over referencing an existing journal's Id in JSON format using the PUT method to the URI referencing the Journal's date. Note that the "Notes" property of the JSON object has been updated to reflect the changes.
5. *Delete an existing Journal:* A DELETE request is sent to the URI of the journal; this time we use the Id of the journal rather than the date to alter the resource.

7 References

[1] FitzMacken, Tom (2013). "*Mocking Entity Framework when Unit Testing ASP.NET Web API 2", http://www.asp.net/web-api/overview/testing-and-debugging/mocking-entity-framework-when-unit-testing-aspnet-web-api-2#dependency*

[2] FitzMacken, Tom (2013). "*Unit Testing ASP.NET Web API 2",* http://www.asp.net/web-api/overview/testing-and-debugging/unit-testing-with-aspnet-web-api#addtoexisting

[3] Richardson, Leonard & Ruby, Sam. "*RESTful Web Services*", Web services for the real world, O'Reilly Media, 454 pages, 2007.

[4] Wasson, Mike (2014). "*Attribute Routing in Web API 2",* http://www.asp.net/web-api/overview/web-api-routing-and-actions/attribute-routing-in-web-api-2

[5] Wasson, Mike (2012). "*Model Validation",* http://www.asp.net/web-api/overview/formats-and-model-binding/model-validation-in-aspnet-web-api

Computer Hardware Experimental Teaching Based on MOOC

Corresponding author QUAN Cheng-bin Email:quancb@tsinghua.edu.cn

CHEN Yong-qiang Email:chenyongqiang@tsinghua.edu.cn

（Department of Computer Science and Technology Tsinghua University, Beijing 100084）

Abstract

There are some shortcomings in current Computer hardware experiment platforms, such as limited functions, low speed, and lack of reliability and stability. These shortcomings are inherent of the design of the platforms. As the needs on teaching experiments increase and computer hardware experiment technologies develop rapidly, these shortcomings have become strong barriers that block the development of computer hardware experiment courses.

Since the old platforms have so many shortcomings, the author proposes a remote computer hardware experiment system after exploring the trend of current experiment platform design development. The paper consists of several parts shown as follows:

1. The author proposes and designs a computer hardware experiment platform based on FPGA and embedded computer. An Embedded Control Model is invented.
2. The author proposes and designs the server application and the client application for the platform. A set of computer hardware experiment operating protocols is proposed.
3. The author proposes and designs series of computer hardware experiment samples based on the experiment system.

4. The functionality, efficiency, reliability, stability of the remote experiment system is analyzed and tested. Four system feature evaluation formulas are obtained.
5. Economic benefit of the experiment system is analyzed and proved. A formula for calculating the engineering benefit is obtained.

Keywords: computer hardware experiment, FPGA, embedded solution, remote experiment system, experiment sample

With large-scale development and popularization of MOOC (massive open online courses), more and more computer software courses have joined the MOOC family. Through online courses, all the students of the world can perform online programming, compiling and executing to check learning results of their own software courses, so as to achieve learning while doing experiment for consolidating. However, the computer hardware courses require a lot of computer hardware support, and students need to perform site operation of computer hardware to achieve complex line connection, chip programming, simulation testing, logical analysis and other steps which can not be achieved in online courses. As a result, it is rather difficult for computer hardware courses to do experiment via online teaching platforms. Accordingly, Department of Computer Science and Technology of Tsinghua University developed a set of network-supporting remote computer hardware experiment platforms. The platform consists of experimental FPGA, control FPGA, network interface, server software system, and WEB Service for client service.

The experimenter can apply for a hardware experimental equipment by using the B/S (browser/server) mode through network, transmit finished hardware design codes to the hardware experiment equipment, a server controls the running of the codes, and the intermediate and final results of running are transmitted in real time back to the computer of the experimenter to perform check analysis on the experiment results, thereby completing the hardware experiment. The platform also supports online remote logic analyzer, remote PS2, USB, VGA input and output and other functions, all of which have been granted with Chinese national patent.

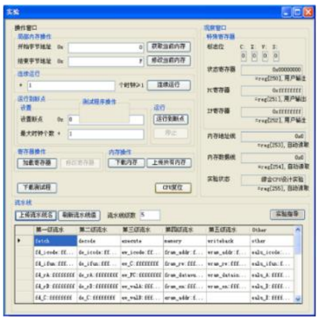
MOOC

Online computer hardware experimental teaching system

Online hardware experimental teaching system consists of experiment FPGA for the experimenter to write in code, control FPGA in charge of system write-in code, controlling on-board circuit, and providing support and monitoring for experimental chip, download CPLD responsible for downloading program to the experimental chip and the control chip, memory for controlling communication transferring, storing related experimental test courses and data, tumbler switch (except that of memory) mounted on a data bus, indicator light and digital tube, USB communication interface and serial port, FLASH memory for the download chip to store download content, various computer hardware experiments, and a server responsible for distribution of experimental platforms.

The experimental platform satisfy not only verification experiment but also open and innovative experiment, and achieve support for experiment of computer hardware courses via network-based hardware platform.

The experimenter writes hardware design codes locally, uploads generated experimental code files which are in file format of RBF to the server through the browser, the server assigns a hardware experiment equipment for separate use of the experimenter, at the same time, the server downloads the codes to the hardware experiment equipment, and controls the experiment FPGA on the experiment equipment to run the uploaded experiment codes by sending one or more clock cycles to the hardware experiment equipment, and performs real-time monitoring on the control FPGA via the experiment FPGA, obtains and stores the intermediate results and final results, and then transmits to the experimenter, and can use the remote logic analyzer of the online experiment platform to carry ou analysis on these data, in order to get logic relationship between hardware running.


实验

操作窗口

局部内存操作

开始字节地址 0x 0 [获取当前内存]

结束字节地址 0x F [修改当前内存]

连续运行

* 1 个时钟>1 [连续运行]

运行到断点

测试程序操作

设置

设置断点 0x 0

最大时钟个数 * 1

运行 [运行到断点] [停止]

寄存器操作 [加载寄存器] [修改寄存器]

内存操作 [下载内存] [上传所有内存]

[下载测试程] [CPU复位]

观察窗口

特殊寄存器

标志位 C: 0 Z: 0 V: 0 S: 0

状态寄存器 0x00000000 =reg[250], 用户输出

PC寄存器 0xffffffff =reg[251], 用户输出

IP寄存器 0xffffffff =reg[252], 用户输出

内存地址线 0x0 =reg[253], 自动读取

内存数据线 0x0 =reg[254], 自动读取

实验状态 综合CPU设计实验 =reg[255], 自动读取

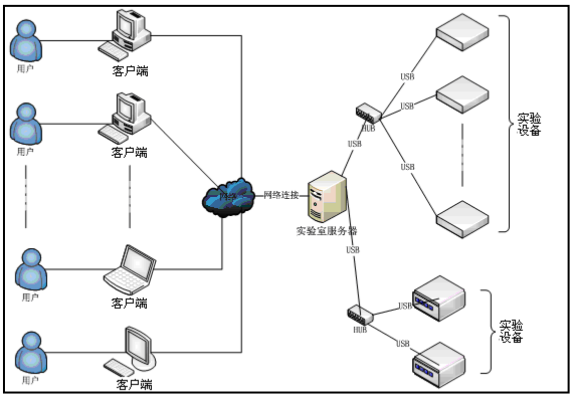
流水线

[上传流水线名] [刷新流水线值] 流水线级数 5 [实验指导]

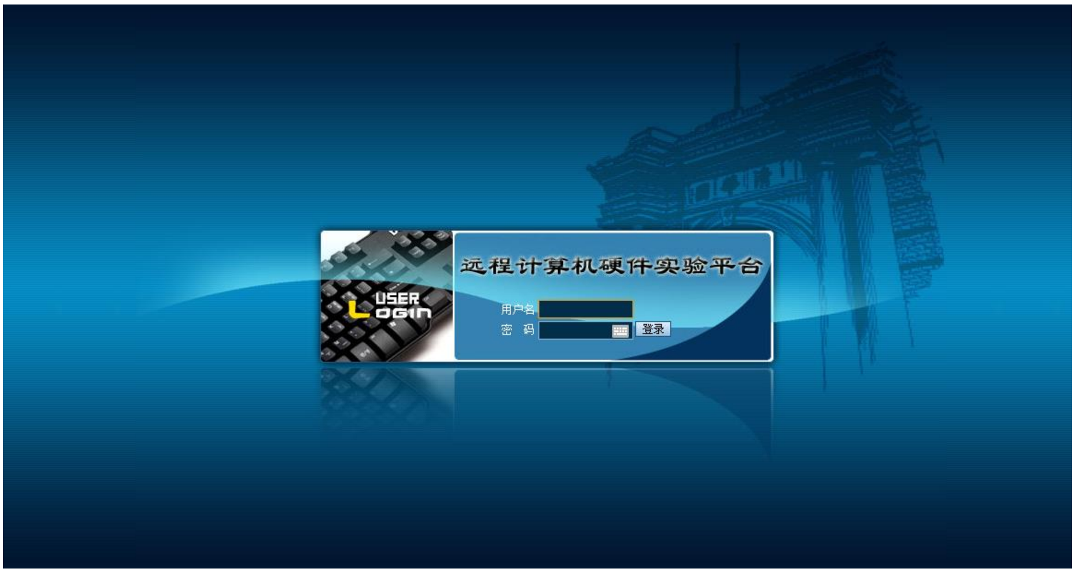
	第一级流水	第二级流水	第三级流水	第四级流水	第五级流水	Other
▸	fetch	decode	execute	memory	writeback	other
	f4_icode:ff...	de_icode:ff...	ew_icode:ff...	fram_addr:f...	wram_addr:f...	valu_icode:...
	f4_ifun:fff...	de_ifun:fff...	ew_C:ffffffff	fram_rw:fff...	wram_rw:fff...	valu_ifun:f...
	f4_rA:ffffffff	de_rA:ffffffff	ew_PC:ffffffff	fram_datavu...	wram_datain...	valu_A:ffff...
	f4_rB:ffffffff	de_rB:ffffffff	ew_valA:fff...	fram_en:fff...	wram_en:fff...	valu_B:ffff...
	f4_C:ffffffff	de_C:ffffffff	ew_valB:fff...	wram_addr:f...		valu_B:ffff...

The experimental interface

Hardware equipment cabinet



Logic diagram of experimental system



Browser Interface

Remote logic analyzer

Logic analyzer at present are needed and hardware directly connected, will be collected by the signal real-time display to the logic analyzer on the screen, which is equivalent to a logic analyzer can only be used or a hardware device, and only to the laboratory for analysis, greatly limits the hardware design of time and space. Remote logic analyzer of the invention will hardware designers from laboratory and hardware equipment, let the designer can be anywhere in the network, as long as the installation of client software, will be able to design their own hardware logical analysis, found the problem, improve their hardware design.

The overall design：

"Signal acquisition FPGA" is the user the experimental operation and experimental data processing terminal. " Transit signal service software " matching service program, responsible for the management of equipment and the user, the user the experiment operation and data transfer. The client is responsible for supporting the login server, client, receive and send signals. The remote logic analyzer structure chart run Arm+linux Control Center.

The control module is embedded ARM solutions, it is a mature module, linux system and the device driver, the designer only needs to application layer in the module of Linux is developed. This module is responsible for collecting signal FPGA and server communication, to establish a communication channel, will be responsible for signal acquisition, real-time transmission equipment operating conditions to the server. The module can be used, if not the module can use USB directly connect to the server computer, it will run the control logic of EPLD signal is forwarded to the server.

The need to store the test program and data in the experiment, we will code via the data bus and address lines written in the memory. SRAM read and write signal is composed of signal acquisition FPGA conversion chip. The SRAM interface signal 5. One is the active low enable EN signal FPGAB_Ram_EN. The second is the active low enable input OE is connected with the signal acquisition, FPGA FPGAB_Ram_OE. The third is to read and write signal RW, connect to the FPGAB_Ram_RW . Fourth , bidirectional data bus Data 16, is connected with the signal acquisition of FPGA FPGAB_Ram_DATA. The 18 bit address bus is connected with the signal acquisition, FPGA FPGAB_Ram_Add. 2 pieces of SRAM using the common address connected to the signal acquisition chip , sending and receiving signal chip to unified signal. Module acquisition signal FPGA, in addition to the address bus, read and write control signals, data bus via acquisition signal chip transit way.

Signal acquisition FPGA Name	Signal acquisition FPGA Attribute	Description
FPGAB_Ram_Add	18bit，out	SRAM address bus
FPGAB_Ram_Data	32bit，inout	SRAM data bus
FPGAB_Ram_EN	1bit，out	SRAM enable signal
FPGAB_Ram_RW	1bit，out	SRAM read and write signal
FPGAB_Ram_OE	1bit，out	SRAM output enable signal

Supported Courses

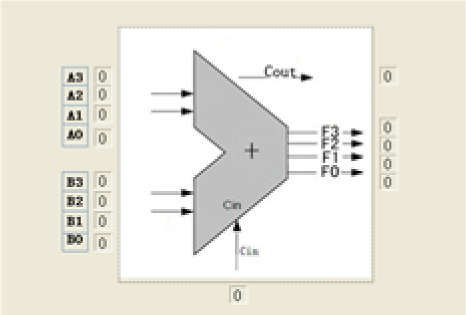
Online computer hardware experimental teaching system supporting the digital logic, computer composition principle of computer courses, system structure, embedded computer and the computer hardware.

Digital logic: half adder, full adder, four bit full adder, encoder, decoder, seven people vote, four responder, BCD-seven segment display decoder, multiplex data selector, register design, frequency divider design, design of 74LS160 counter, eight bits of seven digital tube dynamic display circuit design, a simple state machine design, sequence detector design.

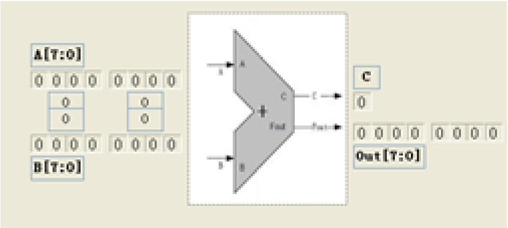
Design simple digital clock.

Principles of computer composition: coding experiments, computing experiment, CPU memory experiment, comprehensive experiment.

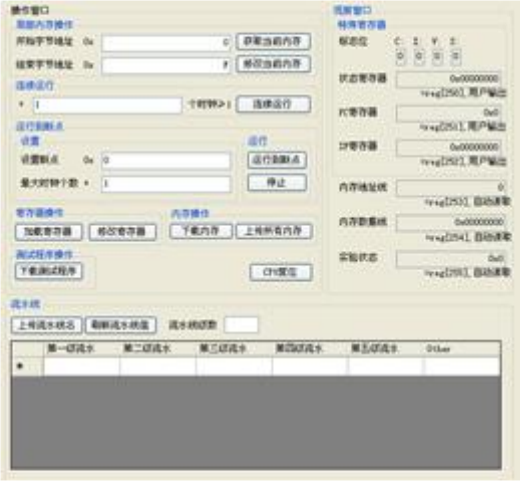
Computer system structure experiment: single cycle CPU experiment, CPU experiment, comprehensive experiment online with CPU, open CACHE processor design.



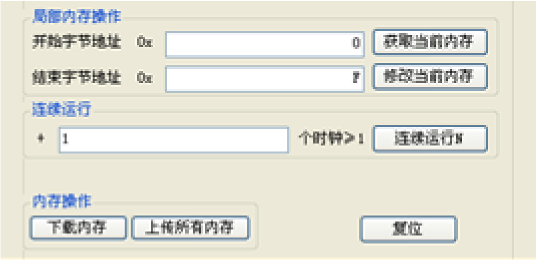
Adder



BCD



CPU With CACHE


局部内存操作

开始字节地址 0x 0

结束字节地址 0x F

连续运行

* 1 个时钟>1

Memory

Effect

The online computer hardware experimental teaching system, learners can in any corner of the world, through the MOOC online education system, apply to join the online teaching of computer hardware, you can apply online to a computer hardware experiment equipment, design their own from the digital logic to the CPU computer hardware, and online debugging, accept feedback the results came back, design compared to their call, remote on-line logic analyzer, modify their hardware design, eventually get the hardware design of the correct. This solves the computer hardware courses must have hardware equipment in the local, to occupy a lot of space, occupy a lot of hardware equipment, and can support 24 hours online, can make computer hardware design.

The remote experiment system, the existing experiment can remote support digital logic, principle, system structure, CPU design course. After the expansion of subsequent development, the system will also support remote experimental interface logic, local support embedded experiment.

Refference

［1］Chen Yongqiang Taopin Embedded Linux transplantation .Research and exploration in laboratory 2012-09-018.

[2] Chen Yongqiang Tang Zhizhong Software and Hardware Design of Embeded System. Research and exploration in laboratory 2011 Feb 2.

[3] Chen Yongqiang Quan Chengbin、Remote Hardware Experiment Teaching.The 2014 World Congress in Computer Science, Computer Engineering, and Applied Computing

（FECS'12：16-19，2012，American）2012

[4]Chen Yongqiang,Quan Chengbin.Design of multi-water Cache CPU based on remote hardware experimental system. Research and exploration in laboratory.10.2012.

SESSION

CURRICULUM AND COURSE DEVELOPMENT + RELATED ISSUES

Chair(s)

TBA

Blending Art and Technology: Two Courses and some MAGIC

Ronald P. Vullo, Ph.D.,
Golisano College of Computing and Information Sciences
Andrew Phelps, M.S., Christopher Egert, Ph.D.,
RIT Center for Media, Arts, Games, Interaction & Creativity (MAGIC)
Rochester Institute of Technology
Rochester, New York 14623

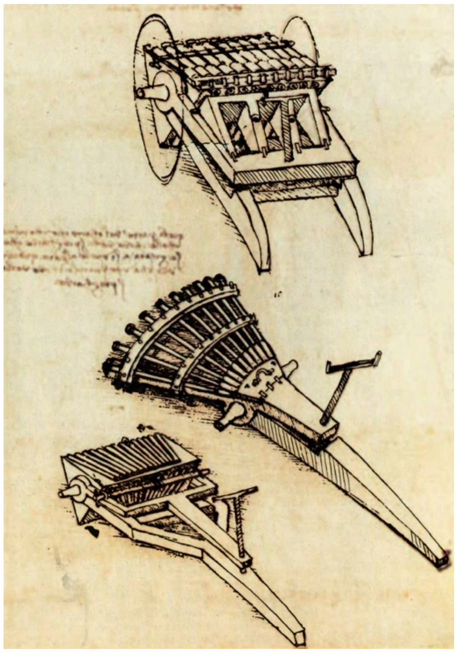
and Heather Rowles, M.A.
Northern Virginia Family Service
Oakton, Virginia 22124

Abstract – *Universities generally separate art and technology, erecting physical and administrative walls between them. This division is artificial and was not always extant. Leonardo Da Vinci found no division between art and technology, and neither do we. We blend them in both the classroom and in our scholarship. Developing technology requires creativity and art requires understanding technology.*

Keywords: Art, Technology, interdisciplinary, curriculum, scholarship, collaborative learning, active learning, STEAM (Science, Technology, Engineering, Art, & Mathematics)

1 Introduction

Mixing Right-Brain and Left-Brain is a longstanding tradition at the Rochester Institute of Technology. After all, we are a polytechnic institute with a world-renowned college of imaging arts and sciences as well as one of the largest and most diverse technical colleges in the United States. Within our faculty ranks within the College of Computing, we have colleagues with a range of backgrounds from Computer Science, Information Sciences, Education, Photography, Painting, Animation, Library Sciences, Medical Sciences, Animation, and Mathematics. Degrees range from the Sciences, Humanities, Education, and the Arts. We also have faculty that have a range of experience from academia, industry, and from entrepreneurial endeavors. So it should come as no surprise that in a field as precise and technical as computing, we believe there is an important place for art and interdisciplinary study. We believe that through a rich intermix of backgrounds and diverse experiences, we are able to show our students the application and societal good that computing can enable and perform. Simply put, we see such mixings of diversity and perspective as the modern-day reinvention of the Da Vinci's Renaissance approach to the sciences and the arts.



There are several contexts in which we can see the interplay between the arts and the sciences within the college of computing. One of these areas is the minor sequence in Web Design and Development, and the other is within RIT's new center for Media, Arts, Games, Interaction, and Creativity (MAGIC).

2 Two Courses

While one of Vullo's goals in creating the minor in Web Design and Development for non-computing majors was to offer technical skills to students in non-technical majors, the other was to bring non-technical majors into the college of computing and foster the interdisciplinary conversations that we believe are essential to a superior education.

Some of these conversations happen in the hallways, some in the college's atrium *Ctrl-Alt-*Deli over sandwiches, and some in the open lab where students work on their projects. But while several of the minor's classes are closed to computing majors, two of them are open to them. These two courses are not counted toward their computing degrees, but are instead considered free electives. Despite that, these courses are popular and enrollment often consists of about a third computing majors. The courses also attract students from such diverse programs on campus as Marketing (E. Philip Saunders College of Business), Psychology (College of Liberal Arts), Media Arts and Technology (School of Print Media), and Professional Photographic Illustration (School of Photographic Arts and Sciences). These two courses are also among the most gender-neutral courses within the college of computing with 50% or greater of female student enrollments. These two classes are Digital Image Creation and Digital Video Creation.

GCCIS-ISTE-205 Digital Image Creation

The focus of this course is the creation and manipulation of digital images to be used for the web. The course covers topics such as basic digital photography, acquisition of images, and intermediate image manipulation.

Students taking this course are exposed to a range of arts and technology perspectives. From the technology side, students learn about the underlying digital concepts related to digital image generation and display. Also, they learn about the technologies required to manipulate images and deploy those images to web-based technologies. Furthermore, they learn about the entire digital image production pipeline and the balance between optimization and presentation within a web-based system.

However, the technology is not the only factor in digital image creation. Students must learn how to take pictures and create content that have impact and contextual appropriateness for their web creations. They must also learn to balance technical optimality with aesthetic experience and quality of the image reproduction.

It is only when the aesthetic and technical competencies combine that the student truly embraces a level of creativity that will allow for new and novel digital experiences.

GCCIS-ISTE-206 Digital Video Creation

This course extends beyond image and moves into the world of video. At a technical level, non-linear video editing includes the fusion of technologies to enable video capture, video manipulation, audio capture, audio manipulation, transitions, effects, and media authoring. It also includes the use of technologies such as video cameras, microphones, mixers, lighting, green-screens, and other related video equipment.

Along with the tools, technological concerns include media formats, compression, resolutions, bit depths, and other concerns that dictate the presentation quality of the media. However, the aesthetics of video creation are also necessary to create compelling stories and media experiences. For example, how to properly frame a scene, how to create audio effects, how to light a scene, and how to set up a composite shot all require a sense of mental composition of the media space. Again, one or the other is not enough, but rather the fusion of these two perspectives is necessary to create the compelling story with video.

Although these courses reside in a college of computing, they are essentially art courses. At first this seems strange, but it falls within the broader multidisciplinary mandate of the curriculum. We do not teach these courses for technology's sake but rather as a domain area that utilizes technology as a medium for expression. Bringing students together from such diverse fields of study presents some challenges, but also makes for a vibrant and exciting classroom environment where students learn and experience both the theory and practice of art and technology. The courses are taught "hands-on" in an Active Learning environment where didactic material is mixed in with a considerable amount of lab work.

In the pre-requisite to these two courses the computing and non-computing majors are taught separately. When coming into these courses from the pre-requisite courses the computing majors are often quite confident based on their technical skills and the non-computing majors are a little bit intimidated. However, it quickly becomes apparent that they each have talents and skills the others will need to learn and develop together. Within the first couple weeks they are working together without giving it a second thought.

3 Some MAGIC

It is this same vision of the value of interdisciplinary collaboration and the blending of art and technology that led Phelps to create the MAGIC (Media Arts Games Interaction & Creativity) Center in collaboration with the RIT Office of the President. It is a conscious and deliberate effort to blur the artificial lines between the arts and the sciences, between technology and expression, between the study of the creation of media and its impact and effect on society and the human condition. MAGIC is a university wide, multi- and cross-disciplinary center in which faculty, staff, and student researchers, artists, and practitioners come together to create, contextualize, and apply new knowledge in a multitude of related fields and disciplines, and as such exists physically and administratively outside of any college at the university.

The MAGIC Center is designed to bridge the gap between research and prototyping, and the ability to bring industry polish and commercial scale and support to myriad projects. This allows these works to have wider and greater impact. For this reason, MAGIC is composed of both a research laboratory (in which many other labs and working groups are housed or affiliated), and a production studio (which leverages our discoveries in ways that broadly disseminates our work to peers and to the public at large). As such, in addition to the academic research, development, and educational side of the MAGIC Center, we have also created the commercial MAGIC Spell Studios, L.L.C. to act as a third-party publisher for student and faculty work. MAGIC Spell Studios is engaged with major distribution networks such as iTunes, Google Play, XBOX Live, etc., essentially providing resources, personnel, and operations for developers in lieu of them having to deal with the complexity and cumbersome nature of setting up their own business.

4 Conclusion

Art and technology are not all that different and are much more intertwined than we often believe. They always have been. When artists develop new paints, that is technology, and when technologists design a language for rendering a user interface onto a screen that is art. There is a need for creativity in technology, as the problem solving employed to enhance a piece of software is, at its core, a creative process. In art, there is also need for technology. Understanding how computers render images is fundamental to creating a beautiful image on a computer. This interweaved nature of art and technology is a reality that we have a tendency to fight in education. Separating creativity and understanding into different colleges in a university or different wings of our elementary schools limits the conversations needed for exploring and developing new and improved art and technologies. We need to encourage students across disciplines to work together. In offering courses that appeal to, and mix, both art and technology students, we can facilitate and inspire conversations that bring and strengthen these domains together.

5 References

Phelps, Andrew et al. http://magic.rit.edu/

Ribauldequins - Leonardo da Vinci studies (http://commons.wikimedia.org/wiki/File:Ribauldequins_-_Leonardo_da_Vinci_studies.jpg).

Leonardo da Vinci - presumed self-portrait (http://en.wikipedia.org/wiki/File:Leonardo_da_Vinci_-_presumed_self-portrait_-_WGA12798.jpg)

Vullo, Ronald P., Ph.D., Catherine I. (Irving) Beaton, M.I.T.E.; A minor in Web Design and Development for Non-computing Majors. Polytechnic Summit 2013, Boston, MA (June, 2013).

Vullo, Ronald P. Ph.D., Catherine I. (Irving) Beaton, M.I.T.E. A "Techy" Minor in Web De-sign and Development for Non-Technical Students. Paper presented at the International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, NV (July 16-19, 2012).

6 About the Authors

Dr. Vullo is Associate Professor, Department of Information Sciences and Technologies, creator and director of the Minor in Web Design and Development for non-computing Majors, and MAGIC Center faculty affiliate.

Mr. Phelps is Professor and Founder, School of Interactive Games and Media, and Director and Founder, MAGIC Center.

Dr. Egert is Associate Professor, School of Interactive Games and Media, and Associate Director, RIT MAGIC Center.

Miss Rowles is an Alumna of RIT’s New Media Information Technology program, Golisano College of Computing and Information Sciences whose interdisciplinary undergraduate work led to her obtaining a Masters degree in Applied History from George Mason University, Fairfax, Virginia

Incorporating Entrepreneurship Topic into a Mobile Computing Program

Xiaoyuan Suo
Department of Math and Computer Science
Webster University
Saint Louis, MO, USA
Xiaoyuansuo51@webster.edu

Abstract— Mobile learning represents exciting new frontiers in education and pedagogy. Webster University has recently started offering courses for Mobile Computing. The designed curriculum focuses on practical experiences, which includes designing, creating, and testing mobile applications on mobile devices. In this paper, we report the curriculum designs, teaching strategies and tools applied; Major feedback also showed students were able to learn programming skills necessary to become more proficient in developing mobile applications.

1. Introduction

Mobile computing has been one of the fastest growing segments of the IT industry and will continue to have a significant impact on how people interact with information and perform their daily activities. A simple search on job recruiting website showed a dramatic increased demand in employment in mobile app development, mobile architecture, mobile design, or other mobile technology related field. A demand of education in mobile technology area has never been higher.

The Department of Computer Science at Webster University has recently started to offer a new B.S. degree in Mobile Computing. Students in our Mobile Computing major focus on applying their technical skills to design, create, and test mobile applications. They will learn skills to be qualified to analyze, design, implement and test mobile applications as well as develop the required skills to maintain and update existing mobile applications.

The unique mobile computing program in Webster University was evaluating existing standard curriculum in an appraisal of the careers on which young programmers embark, and the ways in which both they, and the companies which employ them, wish to develop their capabilities. In the main, findings from the evaluation indicated that students were graduating with a traditional software programming based theoretical degree, which reduced their flexibility to work in different areas of the organization. This resulted in the program approaching the department of management to design a module for their students, which would encourage lateral thinking and the development of initiative. Other reasons, perhaps more fundamental, included the development of skills and knowledge base, to create improved job opportunities, and allow individuals with ideas to transfer them into viable projects, an emphasis on creating market compatible apps.

Clearly, integrating this wide range of aspects and topics in a subject on entrepreneurship for computer science students requires contributions from several disciplines: engineering, computer science sciences, management sciences, marketing knowledge and communications. Entrepreneurship is not new, and during the past number of years it has gained increased interest and research. This increased interest has emerged for a number of reasons, namely the recognition of the contribution of the small firm sector to economic development and job creation. Much of the economy's, ability to innovate, diversify, and create new jobs comes from the small business sector.

Entrepreneurship education has been intensified in universities during the past four decades. In recent years fostering entrepreneurship has become a topic of highest priority in collegiate education. Entrepreneurship, development of personal skills, social and economical aspects of computer technology are of increasing concern for programmers and therefore also for computer and mobile computing education.

This paper will discuss some initial effort of course development, assessment, lessons learned in incorporating entrepreneurship content to the mobile computing program. Detailed data and future work will be provided.

2. Background

Mobility's potential in recruitment is strong. Technology companies are aggressively bolstering their mobile services, with businesses generating 466,000 new jobs in mobile application development since 2007.[1] Mobility has also infiltrated career fairs and is a major focus of recruiters [2]. Mobile learning represents exciting new frontiers in education and pedagogy. Although the mobile systems market is large and the career opportunities for students are excellent, relatively few universities offer mobile computing courses, much less teach mobile systems programming.[3]

The following are the most commonly cited objectives of entrepreneurship education and training programs:

- to acquire knowledge germane to entrepreneurship;
- to acquire skills in the use of techniques, in the analysis of business situations, and in the synthesis of action plans;
- to identify and stimulate entrepreneurial drive, talent and skills;
- to undo the risk-adverse bias of many analytical techniques;
- to develop empathy and support for all unique aspects of entrepreneurship;
- to devise attitudes towards change;
- to encourage new start-ups and other entrepreneurial ventures. [4]

Analysis suggests that a policy aimed at encouraging science-based entrepreneurship should focus on strengthening individual incentives for human capital investment and entrepreneurial behavior both within universities and in business. Key policy areas include attractive tax rates on entrepreneurial income, a tax structure that is not overly progressive, reasonably deregulated labor markets, and a university system characterized by decentralization and competition [5].

Entrepreneurship education has been intensified in universities during the past four decades. In recent years fostering entrepreneurship has become a topic of highest priority in public policy. This trend is due to the widespread recognition that business start-ups are a driving force of economic growth and significant job creation [6]

Entrepreneurship, development of personal skills, social and economical aspects of technology are of increasing concern for programmers and therefore also for computer education.

One of the implications of the shift towards business units is that unit management requires a larger involvement of more people in running a company and a shift from top-down in the direction of bottom-up. Engineers and scientists could no longer focus only on engineering activities, but also have to deal with ‘make or buy’ decisions, marketing, finance, and managerial affairs. This calls for other skills in addition to the regular engineering skills.

Nowadays, science-based entrepreneurship looms large in the public policy arena. This is quite natural given the recent ’scientification’ of technology; in particular, the most rapidly growing and wealth-creating industries such as biotechnology, computers and telecommunications are progressively more science based. But why does such entrepreneurship flourish in some countries, especially the United States, while there seems to be so much less of it in other countries? The purpose of this paper is to attempt to identify some key institutional factors that are crucial determinants of science-based entrepreneurship. Particular attention will be paid to one important subset of science-based entrepreneurship, namely, academic entrepreneurship, This involves the variety of ways in which academics go beyond the production of potentially useful knowledge and take some sort of leadership role in ensuring successful commercialization.

The first strategic choice facing an individual takes place in high school when the young individual decides whether to enter the labor market or to proceed to the university. Given that the individual enrolls at the university, there is a choice between science and technology-based disciplines and other areas, notably the social sciences. At the point of graduation the natural science graduate can again choose between employment and graduate studies with the aim of getting a Ph. D.

Keeping in mind that we focus on the emergence of science-based entrepreneurial ventures such ventures are highly dependent on academically trained and motivated individuals. When talking about academic entrepreneurship, one primarily thinks about university faculty taking an active entrepreneurial and ownership role in these ventures. However, focusing exclusively on this connection is not justified. In addition to university faculty, there are several other important sources for recruiting people to science-based entrepreneurship: From the pool of individuals with either a graduate or an undergraduate exam, and from individuals with that educational background working in other firms.

The study illustrated that entrepreneurship programs are a source of trigger-events, which inspire students (arouse emotions and change mindsets). Inspiration is the program-derived benefit that raises entrepreneurial attitudes and intentions. The insignificant relationship between intention and nascence at the end of the program could be attributed to the well-documented time lag between entrepreneurial intention and behavior, especially concerning undergraduates. [7]

New interdisciplinary programs use faculty teams to develop programs for the nonbusiness students, and there is a growing trend in courses specifically designed for art, engineering, and science students. In addition to courses focused on preparing future entrepreneurs, instructional methodologies are being developed for those who manage entrepreneurs in organizations; potential resource people (accountants, lawyers, and consultants) used by entrepreneurs; and top managers who provide vision and leadership for corporations that must innovate in order to survive [8].

Analysis suggests that a policy aimed at encouraging science-based entrepreneurship should focus on strengthening individual incentives for human capital investment and entrepreneurial behavior both within universities and in business. Key policy areas include attractive tax rates on entrepreneurial income, a tax structure that is not overly progressive, reasonably deregulated labor markets, and a university system characterized by decentralization and competition [5].

Entrepreneurship education and training programs are frequently of very short duration compared to other educational programs concerned with helping people embark on a major career. Some researchers, such as Gibb[17], Sym and Lewis[18] and Curran and Stanworth[7], have found that most small business entrepreneurship programs last as little as a few days, though a few extend over longer periods.

Globalization has given rise to design-produce-service engines with electronically linked elements located across the globe. These virtual entities provide the capacity to produce a wide range of products and services at high quality levels and low costs, but rely heavily on a high volume and steady flow of innovative new concepts to feed the front end of these engines. [9]

3.Program Framework

The program was initially designed for students who want to pursue a degree in mobile computing, mobile app designs, or mobile development. The program was technical based.

3.1 Program learning outcome

After a careful review of some fundamental requirements of a traditional computer science and information technology program, and proposed the following learning outcomes for our unique mobile computing program:

1. Demonstrate critical thinking skills in the field of computer science.
2. Demonstrate the ability to solve problems related to the program content.
3. Demonstrate an understanding of the concepts and principles of software systems.
4. Analyze, design and document a system component using appropriate computer science techniques and models.
5. Make a formal presentation of a software system project including the demonstration of a working application.
6. Demonstrate an in-depth knowledge of advanced mobile computing and development techniques.

The program was proposed in early 2012, and approved later that year. Three introductory to mobile courses in various levels were offered in the first semester (Fall 2012). Program reviews and feedback collected showed students were able to acquire technical aspects of mobile computing. However, some constructive feedbacks were:

- Many students mentioned they need guidance in terms of mobile marketing
- Some needed specific industrial examples of how to market/advertise mobile apps
- More students are interested in the possibilities of working for a mobile app development company. Some even mentioned they want to lead a group of developers.

After a careful review of all comments and the existing courses contents, we made an initial effort of incorporating entrepreneurship content to an existing introduction to mobile computing course. Our effort is supported by the Coleman Fellowship Program, which intend to help university professors to advance entrepreneurship education across campuses.

This pilot course is offered to undergraduate students. The targeted audiences were mostly mobile computing or computer science major; but soon it attracted a wider range of students with different background. Currently the course is offered in an eight-week format as an elective to all students with different background. Enrollment is approximately 15-18 each session. The course not only focuses on teaching students fundamentals of mobile computing techniques, but also has topics on mobile market and mobile entrepreneurship.

Specifically, the course aims to:

1. Identify opportunities that exist in the mobile technology marketplace
2. Demonstrate the tools available to build mobile applications
3. Discover and utilize existing development environments for mobile technology
4. Explain the marketplace for the leading mobile devices
5. Develop a plan of action to start a mobile technology business.

3.2 Course content

In the beginning, the course focuses primarily on mobile technology and computer software programming skills. Students with less computing and mathematics background mentioned they have a difficult time connecting technology terms with market; while students in technology related field asked if there is a professional in the business field who can offer them more guidance in the app market.

Each semester, the course begins with an introductory module creating awareness and understanding of entrepreneurship. It provides the very necessary theory which assist as in the successful development of a new business. This class consists of non-business students, which allows the instructors to customize the module to meet the students' specific needs in terms of prior knowledge and exposure to entrepreneurship.

The second module consists of action learning, where students apply in a very practical basis the knowledge that they have learned in module one. During the module, students develop ideas and project plans. This generates valuable learning experience for all students, as it provides them with an insight into how the various functional areas of a business procedure would benefit their software development processes. A new topic on mobile computing will be introduced in each week. Mobile marketing will be introduced briefly during the first lecture session. Topics include but not limited to: mobile platforms and targeted market, mobile app deployment, mobile developer's initial costs. Invited lecture on mobile entrepreneurship takes place during the seventh week. The lecture consists of introduction, discussions, and students' questions. To keep the students apace with the progress of the lectures, they are urged to discuss their application of the topic at the next lecture.

During the eight weeks of the course, the students met for a nominal four hours of instruction per week during a regularly scheduled class meeting time. In addition to using this time to convey logistic information, the instructors of the course also sought to introduce some of the important background material. These meetings will be described below.

Lectures were delivered to the class on the following topics:
_Financial Reporting and Accounting—Lectures offered by a course instructor as well as by a guest instructor from either Webster University faculty or local companies.

- New mobile product development—Founder/CTO of a local mobile computing start-up company.
- Factors in developing mobile applications, Development strategies—Course faculty.
- Mobile business models—Entrepreneur, professor of entrepreneurship, Webster University Faculty.
- Mobile web and Mobile applications—Course faculty
- Mobile information architecture, Communications via network and the web —Course faculty.
- Device testing—Course faculty
- Future of mobile—Mobile market and Mobile technology Entrepreneurial Plan--Local entrepreneurs, course faculty

An example of a week of teaching plan is shown below:

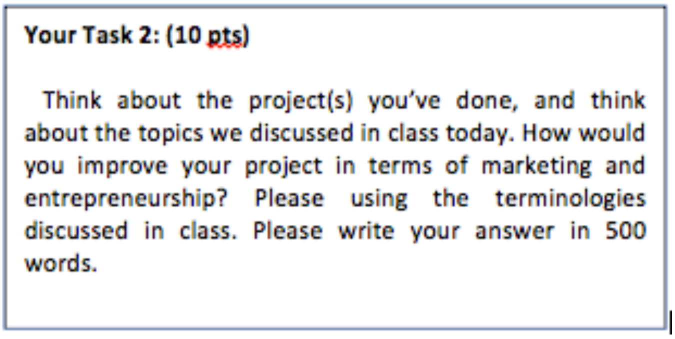
Week 7	Topics: • Putting It All Together • Future of mobile • Mobile market • Mobile technology Entrepreneurial Plan Quiz Assignment

3.3 Course assessment

Students clearly showed tremendous interests in entrepreneurship topics. Questions and discussions carried out during and after the invited guest lectures. Module assessment is conducted based on students' assignment, in class discussion and projects. With a few rare exceptions, majority of the students were able to complete both a satisfactory project and a set of assignment within a week. More than half of the students showed interests in taking additional business and entrepreneurship courses in the future. A few inquired about possibilities of deploying their in-class project as an app to the market.

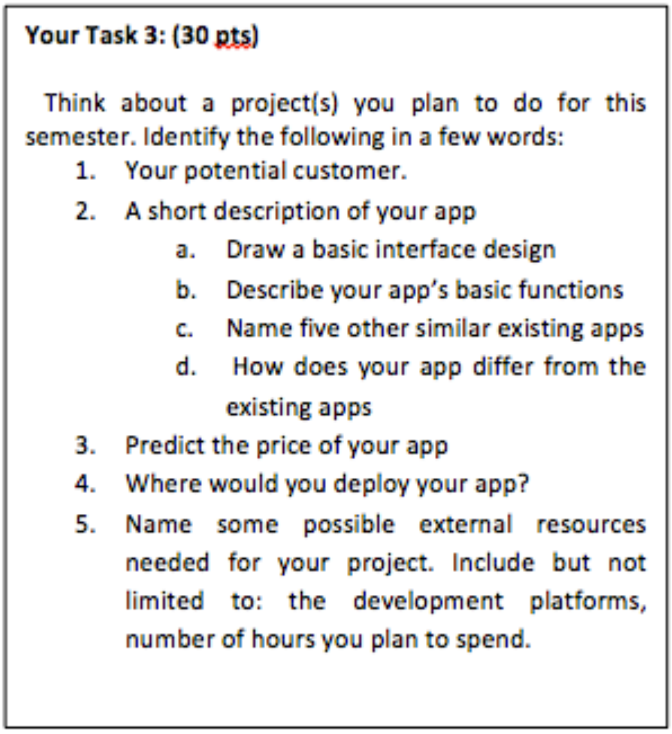
Weekly assignments are project based. Students are required to complete a term project during an eight-week term. After the seventh week's lecture on mobile entrepreneurship, students will be asked to submit a short report on how entrepreneurship topics affect their project design and deployment. A paragraph of a sample assignment is shown below:

A paragraph of a sample assignment from the first week is shown below:



Your Task 2: (10 pts)

Think about the project(s) you've done, and think about the topics we discussed in class today. How would you improve your project in terms of marketing and entrepreneurship? Please using the terminologies discussed in class. Please write your answer in 500 words.



Your Task 3: (30 pts)

Think about a project(s) you plan to do for this semester. Identify the following in a few words:

1. Your potential customer.
2. A short description of your app
   a. Draw a basic interface design
   b. Describe your app's basic functions
   c. Name five other similar existing apps
   d. How does your app differ from the existing apps
3. Predict the price of your app
4. Where would you deploy your app?
5. Name some possible external resources needed for your project. Include but not limited to: the development platforms, number of hours you plan to spend.

Next to the project results (business plans and presentations), it is interesting to focus on the learning results of the students. Clearly, different kinds of learning results can be expected.

All past participants of the course have felt that the integration of the computer science and non-computer science students into a business team is an essential part of the learning experience for both parts of the group. It is not the case that the computer science students carry all of the technical aspects and the non-computer science students the business side of the projects; there is significant crossover in both directions.

We believe, in a long term, this work will advance the understanding of engaging students into new degree programs, in order to prepare students for today's rapidly growing and changing computer technological jobs. The teaching materials and methodologies developed will help improve the students' problem solving skills. The project and the research will generate data and materials that can be beneficial to other universities with a similar degree program.

There were also some practical considerations. One such consideration was that offering a large number of lectures on the subject would take more space in the curriculum than was available. Furthermore, the existing internship in a computer software company was primarily devoted to doing research for the company. Past practice had shown that describing and analyzing the organization would, in general, come down to copying existing organization schemes without any reflection.

In general, students consider it more difficult than expected due to, for instance, the economic viewpoint and the multidisciplinary nature, but they have accepted the subject and learned to deal with it. The course modules as well as the mobile computing education were new to all departments of the university; a number of issues arose which affected the implementation of the teaching materials and methods. The main issues related to the non-computer science students having difficulties in adjusting to the less numeric and less precise descriptive subject areas. Lack of resources and facilities caused problems at the early stages as they curtailed student progression with projects.

4 Conclusion

The program is still under refinement, in the near future, we hope to address all issues sequentially therefore carry out a more successful program. In the near future, we hope to incorporate the content of entrepreneurship to the entire program of mobile computing. More specifically, we hope this effort can benefit fellow educators in the field of computer science.

5 References

[1] M. Mandel. (2012). *The app economy*.

[2] M. Green and M. Perras, "Mobile computing research and education: bridging the gap between academia and industry," in *CASCON '10 Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, Riverton, NJ, USA, 2010.

[3] L. Kulik, "Mobile Computing Systems Programming: A Graduate Distributed Computing Course," *IEEE Distributed Systems Online,* vol. 8, p. 5, 2007.

[4] T. N. Garavan and B. O'Cinneide, "Entrepreneurship Education and Training Programmes: A Review and Evaluation – Part 1," *Journal of European Industrial Training,* vol. 18, p. 10, 1994.

[5] M. Henrekson and N. Rosenberg, "Designing Efficient Institutions for Science-Based Entrepreneurship: Lessons from the US and Sweden " *SSE/EFI Working Paper Series in Economics and Finance,* vol. 410, 2000.

[6] C. Lüthje and N. Franke, "Fostering entrepreneurship through university education and training: Lessons from Massachusetts Institute of Technology," presented at the 2nd Annual Conference on: Innovative Research in Management, Stokholm, Sweden, 2002.

[7] V. Souitaris, S. Zerbinati, and A. Al-Laham, "Do entrepreneurship programmes raise entrepreneurial intention of science and engineering students? The effect of learning, inspiration and resources," *Journal of Business Venturing,* vol. 22, p. 27, 2007.

[8] Z. Block and S. A. Stumpf, *Entrepreneurship education research: experience and challenge*: Center for Entrepreneurial Studies, New York

University, Leonard N. Stern School of Business, 1990.

[9] M. Polczynski and S. Jaskolski, "Entrepreneurial Engineering Education," in *The NCIIA 9th Annual Meeting*, San Diego, CA, 2005.

A Loosely-Coupled Approach to Interdisciplinary Computer Science Education

Thomas Way[1], **Seth Whidden**[2]
[1] Department of Computing Sciences, Villanova University, Villanova, PA, USA
[2] Department of Romance Languages & Literature, Villanova University, Villanova, PA, USA

Abstract - *Interdisciplinary education seems to be on the lips of every university president, dean, provost, department chair, professor, student and industry professional these days. The concept of combining disciplines in a way that students will see them in the real world is appealing and valuable. Implementing a truly interdisciplinary course, where two or more disciplines are merged into a single course, can be an exciting proposition. This endeavor can be time-consuming, labor intensive, cumbersome and administratively challenging. In this paper, a less onerous approach is presented that overcomes many of the hurdles typically encountered in getting an interdisciplinary course offering on its feet. As an initial attempt, Computer Science and French disciplines were successful combined by holding two, simultaneously offered, independent courses that had numerous collaborative lessons and projects designed into them. Quantitative and qualitative results are presented, a realistic analysis of outcomes is given, and guidelines for developing a similar offering are proposed.*

Keywords: Interdisciplinary courses, distributed expertise, machine translation, natural language processing, writing, stylistics.

1 Introduction

The breathtaking pace of change in computing and technology, and its widespread adoption in virtually every human endeavor, has led us to the dawn of a never before seen era of interdisciplinarity. No longer is specialization enough. Nearly all fields of human activity require an understanding and application of that field within the context of one or more other fields. Within education, interdisciplinarity is the combining of two or more disciplines into a single, cross-discipline learning experience. [7]

Beyond academia, interdisciplinarity involves the crossing of traditional topical boundaries in business, science, industry in general and across society [7]. In some cases, these new combinations are inherent, as when a new program of study or research arises, such as nanotechnology which combines ideas from physics, chemistry, and mechanical and electrical engineering [9].

There are a variety of well-recognized benefits and barriers to interdisciplinarity within the academic world, such as the siloed culture of specialization, differences between approaches to teaching and learning among disciplines, differences of opinion regarding the importance of interdisciplinarity, and a host of institutional, administrative and logistic issues. [1, 4, 7]

Traditional approaches to interdisciplinary education frequently involve a true merging of subject areas into a single course. Two, and sometimes more, faculty members collaborate to contribute content and merge material, crafting the course that presents this subject matter in a way that crosses the boundaries of the involved disciplines to produce a new and possibly never-before-offered course. Initial excitement among faculty and administrators can sometimes transform when the reality of creating and offering such a course sets in. Faculty find the effort fulfilling, but also very labor intensive, time consuming and cumbersome. Administrators find the idea compelling and promotable, but wrestle with how to fit the course into rigidly defined course categories, justify extra expenditures on faculty time and materials, and assign course teaching credit to multiple faculty who are teaching the same course, often with lower than typical enrollment.

For classification purposes, we have dubbed the collection of variations on the approaches to creating a merged-topic, high-overhead course the *tightly-coupled interdisciplinary* course approach. This name draws its inspiration from the concept of tightly-coupled parallel processing, where parallel tasks are highly dependent on each other. With a tightly-couple approach, the topics in an interdisciplinary course are highly dependent on each other as they are integrated through careful design and significant effort, to produce what is essentially a brand new discipline that contains merged elements of all original disciplines.

In this paper, we introduce the *loosely-coupled interdisciplinary* approach that is much lighter weight and overcomes many of the organizational and administrative challenges inherent with the traditional, tightly-coupled approach while still providing students with much of the same cross-disciplinary benefits that are gained with a tightly coupled course. The approach decouples much of the dependencies between the topic areas being taught, much in the way a loosely-coupled approach to parallel processing decouples individual tasks.

2 Interdisciplinary Approaches

Significant effort has been invested in a wide variety of interdisciplinary Computer Science teaching approaches [1, 4, 7, 10, 11, 13], and is an active and ongoing area of pedagogical research. These efforts have primarily focused on variations of the tightly-coupled interdisciplinary model, as most work aims at designing a single course and this is the most natural way to think about collaborating on an interdisciplinary project.

The approach of *distributed expertise* (Figure 1) is a form of collaborative education that does not necessarily require crossing disciplinary boundaries, although it provides a model that is applicable. With distributed expertise, faculty members contribute from their own backgrounds in a collaborative effort that draws proportionally from these backgrounds in support of the subject matter. It is a model that can be helpful when forming collaborations across disciplines in the early stages of designing an interdisciplinary course. [3, 11]

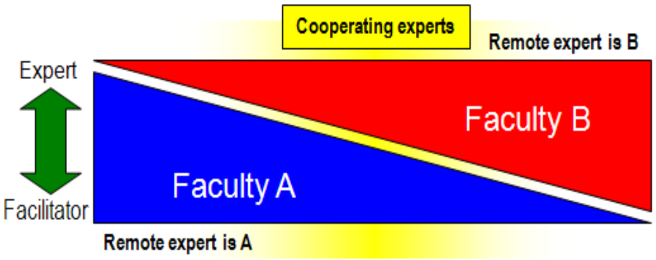


Figure 1. Digital distributed classroom expert-facilitator model continuum.

Team-teaching is an obvious approach to use when developing a course that draws material from disparate disciplines. Using a team-teaching model where an educator from computer science collaborates with an educator from another discipline has been successfully applied to a Project Based Learning (PBL) approach at the high school level [5].

There is evidence that having students collaborate within interdisciplinary project groups is very beneficial to learning at the university level and provides students with significant subject matter relevance [6]. While these approaches can be quite effective, they do not themselves overcome the administrative challenges of accounting for faculty effort and assigning course hour credit equitably in team taught courses at the university level.

While summarizing the vast breadth of experiences with tightly-coupled approaches that are reported in the literature is beyond the scope of this paper, there is great value in identifying the most successful aspects of this approach and finding ways to apply those aspects in the most efficient and effective way possible. Mining the wealth of interdisciplinary course design experience of our colleagues has been an invaluable component of designing the loosely-coupled interdisciplinary model that is presented in this paper.

3 Loosely-Coupled Approach

Given the increasing value of Computer Science to other disciplines [6], the goal of this research is to develop an interdisciplinary instructional approach that brings together students and faculty in two traditionally non-overlapping academic disciplines, one of which is Computer Science, and to do so in a way that minimizes the challenges and maximizes the chances for successful learning.

3.1 Motivation

As computing and technology use has become ubiquitous, the skill of computational thinking has become a valued and necessary intellectual commodity as a literate member of society [12]. Thus, Computer Science educators are positioned ideally to facilitate interdisciplinary courses that put computing in context [1, 2] and provide a relevant and foundational expertise that informs computational thinking across virtually all disciplines, including the arts [8].

Experience with previous efforts to create and offer interdisciplinary course models [1, 10, 11, 13] led to an understanding of many of the difficulties inherent in the traditional, tightly-coupled approaches. Thus, identifying the challenges and ways to overcome them motivated the design of a loosely-coupled interdisciplinary approach to course design.

We recognized the value of the distributed expertise approach and the strong desire at many levels of academia to provide interdisciplinary education opportunities for students and faculty. For any interdisciplinary course design approach to be reusable, it should be easily replicable and attempt to reduce the cumbersomeness that can occur with the tightly-coupled model. A goal perhaps specific to interdisciplinary Computer Sciences courses is to give students a taste of real-world problems and solutions, putting each disciplinary topic within a relevant context of the other disciplinary topics.

3.2 Model

Given the motivations we identified, the concept of the loosely-coupled interdisciplinary approach was designed. Within an interdisciplinary course, there are at least two topic areas that are combined to produce the material covered by the course. In Figures 2 and 3, these two topic areas are labeled “A” and “B”. For the models shown here, these labels can indicate two distinct topics, or individual faculty members, or even two pre-existing courses.

The tightly-coupled model (Figure 2) merges topics A and B into a new topic C. Much effort and time is then invested in devising a semester-long course (indicated by the long arrow line) that thoroughly covers topic C. This diagram also illustrates the investment of time and effort by two faculty members, A and B, in preparing and presenting the topic. The diagram also shows that students are joined in attending class C, and participate together for the duration of the semester is studying topic C.

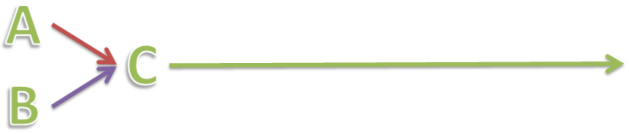


Figure 2. Tightly-coupled model.

The loosely-coupled model (Figure 3), by contrast, does not produce a brand new topic C. Instead, it remains as two distinct topics, A and B, with multiple topical "merge points" during the semester. The diagram demonstrates how two previously existing courses, A and B, taught by two faculty members serving two distinct populations of students, can produce the effect of an interdisciplinary learning experience is a much less cumbersome way.

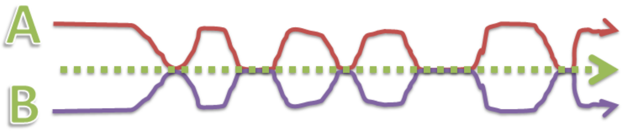


Figure 3. Loosely-coupled model.

In this new approach to interdisciplinary course design, students attend two separate classes taught by two different faculty members, but come together numerous times during a semester to join forces in learning and activities. To apply this model, two interested faculty members first recognize that there can be a meaningful collaboration between the topics of their existing courses, then identify and prepare for multiple points during a typical semester to conduct interdisciplinary activities, have their courses scheduled at the same days and times, and offer their slightly modified courses to their typical student populations.

Risk is reduced, since even if everything went wrong, two courses could run independently impact only to some of the specific material covered. Majors and minors still fulfill their requirements, and aside from disappointment that the interdisciplinary aspects were not explored, everybody is okay

3.3 Course Design

Having devised a model for a lighter-weight approach to interdisciplinary teaching, we implemented the model to combine two discipline-specific courses. To test the model, we prepared and offered two simultaneous undergraduate courses, held at the same class meeting times in adjacent classrooms and with regularly scheduled, combined meetings. These courses were "Machine Translation" offered as an elective course to Computer Science (CS) majors and minors, and "Writing and Stylistics and French" that is a required course for French and Francophone Studies (FFS) majors and an elective for minors. The only administrative issues we dealt with were getting approvals from department chairs in our respective disciplines to combine occasional class meetings and requesting that the Registrar schedule our meetings on the same days and times in nearby classrooms. Informally, we also received enthusiastic support on this approach from our students, colleagues, dean and university president.

Our goal for this structure was to deepen and enrich the educational experiences of all involved through interdisciplinary projects, lectures, activities and discussions at the intersection of human language and its manipulation by computers. The overarching goals were to develop a portable approach for combining CS topics with disciplines that traditionally might be considered non-overlapping with or disjoint from CS; and to extend the FFS curriculum beyond its traditional confines, bringing the study of language to bear on other fields of knowledge. In addition, we hoped that the semester's work would be consistent with the broader missions of our college and university of fostering interdisciplinary learning opportunities.

3.4 Preparation

In order to prepare for our loosely-coupled interdisciplinary teaching experiment, we met a number of times during the summer leading up to the fall semester in which the courses were to be offered. The outcomes of these meetings, which continued throughout the semester in which the courses were offered, included:

- Simultaneous course scheduling - arranging with the university registrar to schedule our courses on the same days and times in nearby classrooms. At least one classroom should be sufficient large to accommodate meetings of all students from both courses.
- Regular solo course preparation - we continued, as in past offerings, to independently prepare our course to assure we covered the necessary material for our major and minor students.
- Planning periodic join-points - we identified approximately 10 topics that our combined group of students would benefit from joint learning, including lectures, activities, discussions and projects.
- Brainstorming student interactions - we invested quite a bit of time in coming up with relevant and engaging ways to have our students interact, usually through

classroom activities, discussions and interdisciplinary projects that required expertise from both disciplines.

- Imagining student projects - Early on, we envisioned that projects would be the key to maximizing student collaboration while maintaining focus on disciplinary knowledge. Software tool-based projects that could analyze French texts of various forms, and that interdisciplinary student teams could create, held promise.

3.5 Implementation

Implementation was done for the Fall 2012 semester and was relatively straightforward, involving an important initial joint class meeting following by both independent, parallel course single-discipline meetings and joint, interdisciplinary meetings. The CS and FFS courses met separately for the first meeting of the semester so that instructors could prepare students for the semester to come. In the next course meeting time, CS and FFS students met together in the same classroom where a joint presentation by the two faculty members was made to introduce the students to the interdisciplinary aspects of the course.

We embraced one notable outcome of previous work with interdisciplinary distributed expertise that found that it was vitally important to have students from different classes who will be collaborating meet each other very early in the semester [11]. Our experience in this offering was that it made a significant difference in how the semester went as students all felt like they knew each other and there was no resistance to collaboration and no academic tribalism as can be seen in approaches that resemble the loosely-coupled concept [11].

Throughout the semester, the courses continued to run independently in regards to the domain-specific content for each course while meeting jointly 10 times for interdisciplinary lectures, discussions and activities.

As the semester progressed, FFS students were required to design a culminating literature analysis project that CS students were required to collaborate on. Interdisciplinary teams of students were formed, with the CS students acting as technical consults to the domain expert FFS students. A number of joint, whole-class brainstorming sessions were held where FFS students presented ideas, CS students proposed solutions and identified aspects that were either too challenging to be accomplished in a single semester or otherwise unworkable, and faculty moderated and guided the process as needed.

Next, CS students invested time designing and implementing tools that sometimes included use of available online tools, reuse of pre-existing solutions being emphasized as an essential part of the process. These solutions were evaluated by FFS students a couple times during the process to provide informal feedback.

During the final two weeks of the course, students met jointly as teams to actively collaborate on using tools that were created by the CS students, modify the tools to produce better results, and generally to interact in a very real-world, interdisciplinary way.

4 Evaluation

The effectiveness of this approach, and the two courses in which we implemented the approach, was measured using student performance on graded assessments, an evaluation instrument that students completed at the beginning and again near the end of the semester, and observations made throughout the semester.

4.1 Student Performance

For the CS course, the average semester grade for the seven students enrolled in the course was A-, with students earning four As, two A-s and one B. All assignments were completed by all students, and the variability in grading was due to thoroughness and consistency of the work done.

The CS course was project-oriented so rather than a final exam students were evaluated on completion of a final project that included significant team-based participation both internally to the CS course and with students in the FFS course. There was significant enthusiasm for the course activities, particularly when they involved the cross-course project work that was done.

For the FFS course, the average semester grade for the eighteen students was B+, with students earning two As, four A-s, six B+s, five Bs and one B-. All students completed all of their assignments, and the variability in grading was due to the differing degrees of preparation for class meetings, for the quantity and quality of participation in French when the FFS class met without our CS partners, and the degree of effort evident in their final project, which was a presentation (in French) of the individual project that each FFS student undertook.

The final papers indicated, in most cases, a high degree of creativity and enthusiasm; the students appreciated the ability to pursue individual projects that were an extension of their existing interests. Every student demonstrated throughout the semester the extent to which this new approach to working with language using computers made them look at familiar materials and an entire field of study in a brand new way. In addition, they developed a deeper appreciation for computers' abilities to analyze different parts of language, and as such they found a greater respect for computer science's problem-solving function.

4.2 Evaluation Instrument

The evaluation instrument measured quantitative and qualitative perceptions of the students regarding various aspects of the combined course material. The four principal, quantitative measures were: skill level with computers in general, computer programming specifically, familiarity with the topic of “Writing and Stylistics in French” and the ways in which computer technology, in particular Machine Translation, can be used in conjunction with the study of Writing and Stylistics in French. Students self-evaluated their ability and familiarity for each of the measures on a 10 point scale, with 0 being lowest and 9 being highest. Table 1 shows the average values for the four categories measured.

Table 1. Average values per class for evaluation instrument measure taken at start and end of Fall 2012 semester.

	CSC 5930		**FRE 1140**	
Categories (0 is lowest, 9 is highest)	**start**	**end**	**start**	**end**
1. Skill level with computers	6.9	8.0	5.0	5.7
2. Computer programming ability	6.1	6.7	1.3	1.3
3. Familiarity with Writing and Stylistics in French	0.3	2.6	5.0	6.9
4. Uses of CS in Writing and Stylistics in French	1.7	5.7	2.8	6.8

Improvement was seen in almost all areas measured (Table 2). Students in the CS course showed slight but measurable improvement in computer skill (1.1) and programming ability (0.6), which was expected and encouraging given that all students were at the junior or senior level and had relatively strong skills at the outset. CS students improved significantly (+2.3) in their familiarity with the subject of the FFS course in spite on only having had a few formal, in class exposures to the topic. This improvement is likely due to the organic exposure to the topic in the course of applying their CS knowledge to solving problems for students in the FFS course, which also led to a marked increase in familiarity (+4.0) with how CS can be used in the study of Writing and Stylistics in French.

One phenomenon that we discovered was in the FFS students’ assessments of their own knowledge of and facility with computers. The short-answer follow-up questions to the numerical indications make clear the fact that the relatively small gains in these two categories reflect not a lack of learning on behalf of the FFS, but rather an inflated sense of what their true abilities were, before they had the opportunity to see the CS students really use computers. This interpretation is supported, we believe, by the increase in familiarity in how CS can be applied to the study of French texts (+4.0). While CS students appeared to have a more realistic view of their own technical capabilities at the outset, there was significant benefit to them gained from applying their science background to solving real and challenging problems from a non-computing discipline.

Table 2. Improvement in evaluation instrument measures per class and combined taken at start and end of Fall 2012 semester.

	Improvement		
Categories (0 is lowest, 9 is highest)	**CSC 5930**	**FRE 1140**	**Overall**
1. Skill level with computers	+1.1	+0.7	+0.9
2. Computer programming ability	+0.6	0.0	+0.3
3. Familiarity with Writing and Stylistics in French	+2.3	+1.9	+2.1
4. Uses of CS in Writing and Stylistics in French	+4.0	+4.0	+4.0

In the qualitative written responses, CS students demonstrated noticeable improvement in understanding of the subject matter of and uses of CS ideas for the Writing and Stylistics in French. These students observed that the experience demonstrated “knowing the limitations of computers can help students understand when or how tools are useful and when they are not” and that there is value in interdisciplinary collaboration because “programming can solve problems in areas that are much different from computer science.” The CS students were particularly satisfied with being able to devise and implement complex software tools that used their programming skills and creativity to assist students in a French course in ways they never imagined prior to this experience.

In their responses to the same questions, FFS students were able to articulate much more clearly how CS can be of significant help in learning about French written texts and their related issues of style. In addition, after seeing French Writing and Stylistics from the CS perspective, students spoke of now being able to see “many things about style and grammar that sometimes go unnoticed.” In addition, they developed an appreciation for (at least some of) “the complexities that come with computer translation programs.” Finally, FFS students felt that each discipline enhanced the other, saying that Machine Translation can help “fully capture the intricacies of a work of literature,” and that “understanding stylistics gives more options and ideas about ways to analyze a text using machines.”

4.3 Observations

Throughout the process of creating and conducting this interdisciplinary endeavor, we observed many benefits to the faculty pedagogical process and to student learning. As faculty, we were able to move outside of our respective specialties to learn about and collaborate on teaching about material from the other’s discipline. We experienced along with our students the excitement of collaboration and mutual discovery, as we observed CS students feeling empowered to offer suggestions and develop solutions using their disciplinary backgrounds and FFS students uncovering their computational thinking skills as they gained deeper understanding of how computer tools could be developed to solve the language analysis problems they wanted to solve.

Students and faculty alike gained a deeper understanding of the importance, in FFS, and other language-oriented disciplines, of keeping an open mind to what constitutes a “text” worthy of study. Indeed, as more and more FFS students are pursuing majors and minors in other fields of study, it becomes increasingly incumbent on FFS as a discipline to think beyond the traditional definitions of literature, in courses when it is appropriate. In FRE 1140, for example, since students were allowed to pursue a project of their own design, there was a wide array of “texts” that they chose, each one worthy of study; a Marketing major chose an advertisement from a marketing campaign; a dual FFS and Political Science student studied passages from Alexis de Tocqueville’s study of early America; and one FFS major chose a French film and its use of subtitles. None of these approaches would have been likely in a traditional Writing and Stylistics course, and so one of the tremendous benefits to learning came from empowering the students to suggest texts to work on.

5 Guidelines

The loosely-coupled interdisciplinary approach enabled us to offer a lightweight, easily implemented interdisciplinary pair of courses. We believe that replicating this approach, while taking a concerted effort, can be done with far fewer of the challenges typically seen with tightly-coupled interdisciplinary approaches. The following checklist of steps to follow comprises our recommended roadmap for the design of a loosely-coupled interdisciplinary course duo that matches Computer Science with another discipline.

For junior faculty eager to apply the latest ideas in interdisciplinarity yet wary of straying from their specialization, we believe this loosely-coupled approach to be ideal. By following this roadmap, a junior faculty’s tenure progress can be protected since their own course in the pair is officially taught is within their own discipline. As a bonus, doing so can simultaneously fulfill departmental, college and university interdisciplinary initiatives or missions.

5.1 Steps to Implement

While there are likely to be as many ways to implement a loosely-couple interdisciplinary model as there are faculty members, the following items summarize what we view as essential steps in this process.

✓ Identify a colleague with whom you have a good rapport. You will spend a lot of time collaborating both outside and inside the classroom, so your styles of preparation and teaching should mesh well.

✓ Identify courses you each teach that could combine well in a loosely-coupled approach. Note aspects of the non-CS topic that lend themselves to a computational solution of some sort.

✓ Determine if there are enough common topic points that your two disciplines. These points are the truly interdisciplinary components of the semester where students from both courses will meet to learn and collaborate.

✓ Meet and sketch out an initial plan for how a semester would flow. Organize topics and meetings within the semester syllabus and schedule to accommodate the loosely-coupled approach.

✓ Seek approval of necessary administrators at the departmental, program, college and university levels. Once you have a plan that works for a course pair that you believe would be a valuable contribution to your institution’s offerings, get the needed approvals.

✓ Arrange to have courses offered at the same days and times nearby each other. The registrar’s office is an invaluable ally in making the loosely-couple model workable.

✓ Promote the collaborative course offerings, encourage students to enroll. Get the word out to whoever is advising students who are eligible for your courses, and take advantage of any opportunities available to promote your unique offering.

✓ Schedule topics to be taught separately and together, coordinate interdisciplinary activities. Once it really is going to happen, spend more time planning out each week of the semester and all of the separate and combined lectures, activities, assignments and projects that will form the interdisciplinary aspects of the courses.

✓ Prepare materials, discussion starters, and activities. Getting out ahead of the schedule as much as possible will enable the more involved synchronization needed versus a typical, single-subject course.

✓ Offer the courses and be prepared for change and adaptation. It is difficult to anticipate exactly how such a unique approach to combining two courses will proceed, so anticipating the need to modify plans on the fly in response to progress will be helpful.

6 Conclusions & Future Work

The design, preparation and offering of this new loosely-coupled approach to offering interdisciplinary courses were very satisfying from a professional and personal point of view. Students were regular and enthusiastic attendees and participants in a wide variety of learning activities, some of which were quite different from activities they had experienced in other courses. Materials were developed and collected so that the course can be offered in the future with a reduced quantity of preparation.

Student output included producing software specifications and usable software tools of potential interest to others in our university community who might work with language analysis, including languages other than French. Learning was significant, as demonstrated by the evaluation instrument, particularly in the area of the interplay between the application of Computer Science technology to the study of Writing and Stylistics in French, which was a primary goal of the course. We feel that this unique combined course offering has demonstrated that a loosely-coupled approach to interdisciplinary teaching can challenge students, facilitate quicker development and offering of such interdisciplinary opportunities for faculty and students, and ultimately lead to greater, more enjoyable, and more relevant learning than a more traditional approach.

We are offering the pairing a second time in the Fall 2014 semester, which will provide an opportunity to compare and improve the approach. The loosely-coupled approach is also being discussed for proposed future course pairings of CS and Criminal Justice (Cybersecurity), CS and History (Big Data Analysis of the Historical Record), CS and Political Science (Analysis of Political Bias in the Media), and CS with English or Spanish in a similar, computational approach to literary analysis. The loosely-coupled approach is being redefined more specifically as a "conjoined courses" approach to more precisely describe the nature of the collaboration, with future evaluation and exploration of the approach expected.

7 Acknowledgments

The work reported in this paper was partially funded by NSF CPATH award IS-0829616, and by an institutional course development grant from the Villanova Institute for Teaching And Learning (VITAL). Thanks to John Kelley and Carol Weiss for the sage advice and guidance in the development, implementation and evaluation of this work.

8 References

[1] Lori Carter. "Interdisciplinary computing classes: worth the effort." SIGCSE Technical Symposium (SIGCSE 2014), pp. 445-450, 2014.

[2] Lillian Cassel, Robert E. Beck and Richard H. Austing, "Interdependence of Disciplines: Computer Science as a Full Partner," Proc. of the 5th World Conference on Computers in Education, 1990.

[3] Lillian N. Cassel and Deepak Kumar, "Distributed Expertise in Teaching," The 7th IFIP World Conference on Networking the Learner: Computers in Education, Copenhagen, Denmark, pp. 881-886, 2001.

[4] Lillian N. Cassel. "Interdisciplinary computing is the answer: now, what was the question?" ACM Inroads, vol. 2, no. 1, pp. 4-6, March 2011.

[5] Ling-Chian Changa and Greg C. Lee. "A team-teaching model for practicing project-based learning in high school: Collaboration between computer and subject teachers." Computers & Education, vol. 55, no. 3, pp. 961-969, November 2010.

[6] Debra S. Goldberg and Elizabeth K. White. "E pluribus, plurima: the synergy of interdisciplinary class groups." SIGCSE Technical Symposium (SIGCSE 2014), pp. 457-462, 2014.

[7] Julie Thompson Klein. Interdisciplinarity: History, Theory, and Practice. Detroit: Wayne State University, 1990.

[8] Bill Manaris, Renee McCauley, Marian Mazzone, and William Bares. "Computing in the arts: a model curriculum." SIGCSE Technical Symposium (SIGCSE 2014), pp. 451-456, 2014.

[9] Alan L. Porter and Jan Youtie. "How interdisciplinary is nanotechnology?" Journal of Nanoparticle Research, vol. 11, no. 5, pp. 1023-1041, 2009.

[10] S. M. Pulimood and U. Wolz, "Problem Solving in Community: A Necessary Shift in CS Pedagogy," SIGCSE Technical Symposium (SIGCSE 2008), pp. 210-214, 2008.

[11] Thomas Way, Lillian Cassel, Kim Pearson, Ursula Wolz, Deborah Tatar and Steve Harrison. "A Distributed Expertise Model for Teaching Computing Across Disciplines and Institutions." The 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 2010), Las Vegas, Nevada, July 12-15, 2010.

[12] Jeanette M. Wing, "Computational Thinking," Communications of the ACM, vol. 49, no. 3, pp. 33-35, March 2006.

[13] Ursula Wolz, C. Ault and T. M. Nakra. "Teaching Game Design through Cross-Disciplinary Content and Individualized Student Deliverables," The Journal of Game Development, 2(1), Feb. 2007.

Methods for Teaching a First Parallel Computing Course to Undergraduate Computer Science Students

Mike Estep, Feridoon Moinian, Johnny Carroll, Chao Zhao
Computing & Technology Department, Cameron University, Lawton, OK, USA

Abstract – *Parallel computing has gained popularity in recent years. It takes advantage of multiprocessor computing equipment, such as multicore processors, computer networks, clusters, and massively parallel processors. Complex problems are solved more efficiently by overcoming the physical constraints of serial computing. The basic idea of parallel computing is to divide a large problem into smaller ones which can be carried out by mapping onto different processors; therefore those subtasks can be performed simultaneously. However, parallel computing instruction for undergraduate students is still somewhat in its infancy and continues to pose certain challenges in terms of resource availability. This article describes current instruction methods, obtaining of valuable instructor and student resources, and observations in teaching a first parallel computing class to undergraduate students at Cameron University*.

Keywords: Teaching, Parallel Computing, MPI, OSCER

1 Introduction

Parallel computing has gained popularity in recent years. It takes advantage of multiprocessor computing equipment, such as multicore processors, computer networks, clusters, and massively parallel processors (MPP). Complex problems are solved more efficiently by overcoming the physical constraints of serial computing. The basic idea of parallel computing is to divide a large problem into smaller ones which can be carried out by mapping onto different processors; therefore those subtasks can be performed simultaneously.

There are two basic parallel computing forms: data parallelism and task parallelism. Data parallelism processes a portion of the entire data set, such as sorting a big array with multiple processes; each process sorts a part of the array, and then processes interchange data such that all the elements held by process p_i are less than or equal to those held by process p_{i+1}. That is, the elements in the entire array are sorted in a non-descending order. Task parallelism focuses on distributing different functions to different processors with the same or different data.

Due to the increasing importance of parallel computing, Parallel Algorithms was one of the recommended courses in the ACM/IEEE CS2008 curriculum recommendation [1]. Furthermore, parallel and distributed computing was listed as one of recommended knowledge areas in the ACM/IEEE CS2013 curriculum recommendations [2]. Hence, it is essential to offer a parallel computing course at the undergraduate level. However it may be very challenging to teach such a class at a university without any parallel computing equipment. In this article, the authors discuss utilizing the Oklahoma Supercomputing Center for Education and Research (OSCER) [3,4] to teach a parallel computing class at Cameron University.

2 Methods

Cameron University is a five-year public regional university that offers a B. S. degree in Computer Science (CS) in the Computing and Technology department. CS 3813 Parallel Computing is one of elective courses in the CS curriculum, usually offered in the spring semester each academic year. Topics in CS 3813 include: parallel algorithms and implementations for sorting, searching, matrix processing and other problems, and efficiency issues of parallel algorithms on different architectures. In order to teach this class efficiently, several methods are utilized.

2.1 Instructor Training

Teaching a parallel computing class can be quite challenging since available instructional resources are relatively limited. To obtain the most current information on parallel computing, the authors attended OSCER’s parallel computing workshop at the University of Oklahoma that is sponsored by the National Science Foundation (NSF). In the workshop, some essential topics in parallel computing were covered, such as parallel logic, basic Message Passing Interface (MPI) and advanced MPI, parallel equipment manufacturing and usage, and MPI programming exercises. Workshop attendants from different academic institutes exchanged ideas over teaching strategies, teaching methods, and teaching content. Through these activities, the authors gained much information and knowledge about parallel computing, useful for class instruction.

2.2 Inviting Parallel Processing Professionals to Visit the Cameron Campus

In order to motivate Cameron University CS student interest in parallel computing, experts were invited to deliver seminars on the subject. Basic topics included, but were not limited to: trends in parallel computing, interprocess communication with memory sharing in a multiple central processing unit (CPU) system, interprocess communication in a distributed system, and job balance in parallel computing efficiency. Meanwhile OSCER offered an online remote account to each CS student over a supercomputer with thousands of processors and massive storage capability. Each student would run their parallel programs on the machine. The OSCER experts made a special trip to Cameron to teach students how to configure and use their accounts [3].

2.3 Visiting the University of Oklahoma Supercomputing Center

Considering that most Cameron students did not have much opportunity to see a real supercomputer, the authors organized a field trip to the Supercomputing Center located at the University of Oklahoma. The director of the computing center gave the students a well-organized tour, and the students obtained a great deal of information, such as supercomputer physical components, architecture, organization, and power supply. This knowledge was useful in helping students run their programs on a supercomputer.

2.4 Parallel Algorithm Development

Parallel algorithm development can be quite difficult for CS undergraduate students who are used to serial programming with single processes. To solve this problem, the authors focused on explaining the major differences between sequential and parallel computing. Sequential computing completes a task step-by-step in a certain order by one process, while parallel computing divides the task into a number of subtasks and then finishes tasks in parallel utilizing different processes. For example, if a search is conducted sequentially using a key item from a very large list, the key must be compared with each item in the list until the first appearance is met, or the end of the list is reached. In a parallel approach, the list can be divided into N sections, and then a section and key are passed to a process, next all processes run simultaneously, and finally execution results are collected by one designated process to reach a conclusion.

Parallel computing can use much less time to complete the same task. That is, if a single process uses time T to complete a task, and n processes may use equal or bigger than time T/n to complete the same task, where n is a positive integer, T/n is less than T if n is equal or larger than 2. This is called the “speed up” of how quick a task can be completed with n processes. The speed up concept can be expressed in the following formula:

$Sp_{(n)} = Ts / Tp$

Ts is the optimized sequential time needed by a single process to complete the task, and Tp is the parallel time needed by n processes running in parallel to complete the same task. For example, if one person can finish a painting job in one hour (Ts), as two (n) persons only use 40 (Tp) minutes to complete the same job, and then the speed up ($Sp_{(n)}$) is 60/40 = 1.5, that is, the job can be completed 1.5 times faster.

However, in general, parallel computing efficiency is less than 100%. Parallel computing efficiency can be defined as:

$Ep_{(n)} = Ts / (Tp * n)$ or
$Ep_{(n)} = Sp_{(n)} / n$

$Ep_{(n)}$ is efficiency, n is the number of processes, $Sp_{(n)}$ is the speed up, Ts is the optimized sequential time needed for a single process, and Tp is the parallel time needed for n processes. From the above example, $Ep_{(n)} = 1.5 / 2 = 0.75$ or $Ep_{(n)} = 60 / (40 * 2) = 60 / 80 = 0.75$. Therefore the efficiency is 75%. In conclusion, a parallel algorithm is faster, but it is also expensive because the interprocess communication always causes some overhead. It is of significant concern to attempt overhead reduction in parallel algorithm development.

Generally speaking, a parallel algorithm may have three parts: a common part, a master process part, and a slave process part. The common part includes necessary variable declaration and initialization, and shared functions declaration and implementation. The master part is designed to divide the given data set into subsections, send a subsection to a slave process, collect the computing results from the slave process if necessary, and finally display the results if required. The slave part is used to complete a subtask that includes: receiving necessary data from the master, completing calculation on the data, and sending the result back to master if required. In summary, a parallel algorithm design could be expressed in the following common function prototypes…

```
main function
{
    variables declaration and initialization;
    if (master)
    {
        code that will be executed by the master
        process;
    }
    else
    {
        code that will be executed by a slave process;
    }
        program termination part;
}
```

2.5 Teaching Methods

In CS 3813, the following topics were covered: parallel computing without communication among slave processes, job balance and dynamic job assignment, parallel computing with communication among slave process, derived data types, and new communicator creation. MPI was used to implement parallel algorithms in C/C++. In order to help students understand and master the above topics, typical examples were given from *Using MPI Portable Parallel Programming with the Message-Passing Interface* (2nd Edition) [5].

2.5.1 Numerical Integration

Numerical integration to compute pi was used to teach parallel computing without communication among slave processes:

Since
$\int_0^1 1/(1+x^2)\,dx = \arctan(x)\,|_0^1 \arctan(1) - \arctan(0) = \pi/4$,
$\pi = 4 * \int_0^1 1/(1+x^2)\,dx$

So that we can integrate the function
$f(x) = 4 / (1 + x^2)$

When
$x = 0$, $f(x) = 4$, and $x = 1$, $f(x) = 2$

The x interval is first divided between 0 and 1 into N sections, each section being a small rectangle with width = 1 /N, and height = $4 / (1 + x^2)$, and then the rectangles are added together to obtain pi's value. To do so, first the master process accepts an N from the keyboard, and then broadcasts it to all processes. Secondly, all the processes receive N, find the width of the small rectangles by 1/N, and then find the mid-point of the width for process p_i using:

mid-point = width * (p_i's rank + 0.5)

For example, if p_i's rank, where p_i's rank is between 0 and N-1, is 1 and N = 10, then the mid-point is 1/10 * 1.5 = 0.15, and therefore the second process will start its calculation at x = 0.15. Each process takes one or more of these small rectangles to compute their areas. Finally after completing a calculation, all processes send the calculation results to a destination process, and then the destination process adds all results together to display pi's value. If the computing result is sufficient enough, the computing process will be stopped; otherwise another round of computing may be conducted as described. It makes sense that the bigger N is, the better pi value will be expected in a certain range. This is a good example because calculation for each section is independent from each other, and therefore these processes can run simultaneously. To enhance student learning, it was given to the students as a programming assignment using n processes to sum a large integer array.

2.5.2 Job Balance

Job balance is an important consideration in developing a parallel algorithm because CPUs involved in parallel computing may have different computing capacities. Therefore some CPUs may be busy all time, while some CPUs may be idle. CPUs are the most valuable computing resource, and it is desirable to keep them as busy as possible. To do so, CPUs should receive a new sub job once the current sub job is finished, until the entire job is completed. That is, it is important to allow faster CPUs to complete more jobs than slower CPUs. Matrix multiplication was used as an example to explain this topic. Matrix multiplication can be expressed as:

A (m, n) * B (m', n') = C (m, n')

where m and m' are the number of rows in A and B and n and n' are the number of columns in A and B respectively, and n must be equal to m'.

Master process performs the following in order:

A. Broadcasting matrix B to all slave processes;
B. Sending a row of matrix A to each process.
C. Receiving a row of matrix C from a slave process.
D. Copying the received row into matrix C
E. If the number of sent rows is less than the number of rows in matrix A, repeat B, C, D, and E until the job is done.

Slave process performs the following in order:

A. Receiving matrix B;
B. Receiving a row r of matrix A;
C. Multiplying row r to matrix B to produce a row of matrix C
D. Sending the resulted row back to the master process
E. Repeating B, C, and D until the completion notice is received.

This algorithm allows "self scheduling", that is, after a process completes its job, another job will be given until the entire job is done. Matrix multiplication was assigned to the students as a programming project.

2.5.3 Communicator Creation

In parallel computing, processes are commonly assigned to a number of groups, and each group completes a specific task. There may be many groups, but a process only belongs to one group. A group of processes with a unique context assigned by the system is called a communicator.

The Monte Carlo method to compute pi was introduced to explain this concept. The Monte Carlo method is a class of computational algorithms that relies on repeated random sampling to compute results. This method may not be the best way to compute pi, but can be an efficient way to show how to create a new communicator. The basic mathematical algorithm is:

If a circle of radius r = 1 is inscribed inside a square with side length 2, then the area of the circle will be $\pi * r^2 = \pi$ and the area of the square will be 2 * 2 = 4. So, the ratio of the area of the circle to the area of the square will be $\pi/4$. This can be expressed as:

ratio = π / 4 such that
π = 4 * ratio

where ratio can be determined by the number of points inside of the circle over the number of points inside of the square. For any point (x, y), if $x^2 + y^2 < r^2$, where x and y are coordinates of the point and r = 1, the point is inside of the circle and square, and otherwise only inside of the square. Therefore ratio is:

ratio = the number of points inside of the circle / (the sum of points in the circle and the square)

Since the calculation for each point is independent, it can be completed perfectly in parallel. There are two different tasks in computing pi: generating a set of random numbers and calculating points from a given random number set. All processes can be assigned into two groups: one group is a server group that is responsible to generate a set of random numbers, and another group calculates points based on the random number set. Then the formula $\pi = 4 * \text{ratio}$ is used to determine pi's value. A communicator can be created for each group. In the MPI, a set of routines are offered to create a new communicator from an existing one. The algorithm is: (1) create a new work group, world group, from the default communicator that includes all the processes, (2) exclude the server process from the world-group to form another work group, worker-group, (3) create a new communicator using the worker-group that includes all the processes except the server process, (4) free the world-group and worker-group. The results are two communicators: a world-communicator (default) that contains all the processes and a worker-communicator that only includes all the worker processes. In the worker-communicator, worker processes cooperate together to complete the calculation. The server process generates a random number set if noticed. The algorithm is shown below:

```
Server process:
    A. generates a set of random numbers
    B. sends the set to worker processes
    C. if job is not done, repeat A and B, otherwise
       terminates.
Worker processes:
    A. receive a set of random numbers,
    B. compute points from the set
    C. determine if the job is complete; if not complete,
       request server to produce another set of random
       numbers, and then repeat A and B; otherwise send
       the server a job complete signal to end the
       calculation
```

The worker communicator contains fewer processes than the world communicator, such that it may save time on collective operations, therefore it may result in a better performance.

2.5.4 Derived Data Types

Derived data types are very important because they provide a useful means to pack related data that could be homogenous or heterogeneous together as a single data unit to avoid multiple data exchanging among processes. In MPI, data will be sent as is. Hence, it is crucial to pack data together properly before sending them. The data type in MPI is an object that consists of a sequence of the basic data types, such as MPI_INT, MPI_CHAR, MPI_FLOAT, MPI_DOUBLE, and displacements of each of these data types. These sequences and displacements can be described in the typemap:

typemap = {(type0, disp0), (type1, disp1), …, (type $_{(n-1),}$ disp $_{(n-1)}$)}

Data type tells MPI how to interpret data when data is sent or received, while displacement tells MPI where to find data bits when sending them or where to store data bits when receiving them. For example, if typemap = {(double 0), (int 8)}, then the low bound that can be considered the location of the first byte described by the data type is 0, and the upper bound that can be considered as the location of the last byte described by the data type is 16 (8 +4 +4), therefore 8 is the number of bytes occupied by the variable with double data type, the first 4 is the length of an integer, and the second 4 is the number of bytes used as padding to meet system alignment. The difference between the lower bound and upper bound is called the extent which defines as the length of one derived data. So the extent is 16 in this instance. In C, one of the most common alignment requirements is that the address of an element in bytes is a multiple of the length of that element in bytes. If there are 4 such elements in a buffer, then double variable addresses will be 0, 16, 32, and 48. MPI provides a set of routines to get the lower bound, upper bound, and extent.

There are two cases: all elements in the derived data type have the same data type and elements have different data types. In the first case, all data have the same size so that there is no need to add padding bytes to meet the system alignment requirements. Therefore, one can simply pack the elements by using an MPI routine MPI_Type_continuous, commit the new data type, and then use it and free it after usage. The idea is shown in the following code segment:

```
typedef struct
{
float x, y;
} point_type;

point_type Points[100];
…
MPI_Type_continuous (2, MPI_FLOAT, &new_type);
MPI_Type_commit (&new_type);
MPI_send (Points, 100, new_type, dest, tage, comm)
…
MPI_Type_free (&new_type);
…
```

In the second case, a derived data type contains different default data types such that one has to determine the number of data types, the number of elements in each data type, and displacement for each data type. Fortunately, MPI offers a set of routines to complete this complex task. This was introduced to students as following:

```
typedef struct
{
        int pid, priority;
        double arrival_time, cpu_time, start_time;
}job_type;

job_type jobs [100]; //declare an array of jobs
int blocks [2] = {2, 4}; //set up 2 blocks
MPI_Datatype types [2];
MPI_Aint displacements [2];
MPI_Datatype jobs_type;
/*initialize displacements */
MPI_Address (&job_type.pid, displacements);
MPI_Address (&job_type.arrival_time, displacement +
                    1);
/*initialize datatypes */
types [0] = MPI_INT;
type [1] = MPI_DOUBLE;
/*make displacements relative */
displacements [1] = displacements[0];
displacements [0] = 0;
MPI_Type_struct (2, blocks, displacements, types,
                          &job_struct);
MPI_Type_commit (&job_struct);
/*use the derived data type just as default data types */
for (int i = 0; i < 100; i ++)
{
        MPI_Send( &jobs + i, 1, job_struct, i+1, tag, comm);
        …..
}
```

The derived data types in parallel may prove more difficult to understand than data types in sequential computing. Hence, it can require more effort and patience on the part of the instructor to help students understand and use derived data types in parallel.

3 Conclusions

From the authors' teaching practices and observations, the following conclusions are drawn: (1) instructor training in parallel computing is essential in ensuring prerequisite knowledge and skills in instruction of CS undergraduate students, (2) OSCER offers valuable resources that can be used to improve the quality of teaching and learning, and (3) Proper teaching methods provide instructors with an efficient way to deliver their teaching content. Further study should be helpful toward the understanding and development of sufficient student learning outcomes.

5 References

[1] **Computer Science Curriculum 2008: An Interim Revision of CS 2001, Report from the Interim Review Task Force, includes update of the CS2001 body of knowledge plus commentary**. December 2008, Association for Computing Machinery IEEE Computer Society http://www.acm.org//education/curricula/ComputerScience2008.pdf.

[2] **Computer Science Curricula 2013, Curriculum Guidelines for Undergraduate Degree Programs in Computer Science**. December 20, 2013, The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society, http://www.acm.org/education/CS2013-final-report.pdf.

[3] **OU Supercomputing Center for Education & Research (OSCER)**. http://www.oscer.ou.edu/.

[4] **Using a Shared, Remote Cluster for Teaching HPC**. C. Carley, K. Larry, B. McKinney, C. Zhao, and H. Neeman, IEEE2013_indianapolis, ISBN: 978-1-4799-0896-7.

[5] **Using MPI Portable Parallel Programming with the Message-Passing Interface (2nd Edition)**. William Group, Ewing Lusk, and Anthony Skjellum, MIT Press, Cambridge, MA.

Web Programming Curriculum: What Changes, What Remains the Same Over Time

R. Byrd, M. Beier
Abilene Christian University, Abilene, Texas, United States

Abstract – *This study investigated the change in web development course curriculum and pedagogies over time. Data were collected from course materials, assignments, and from student interviews over three years. One of the key differences identified between web development and traditional programming language courses was the complexity of learning multiple languages and technologies within the same course and even with in the same project assignment. Additionally, it was found that an effective web development curriculum requires students to understand more than how to solve a programming problem. Effective web development curricula expose students to the various web technologies, including server configuration, security, networking, employing databases to create interactive content, and human-computer interface optimization. Skills and techniques found to be critical in this study were the ability to simultaneously manage multiple working environments, application aspects, and current technologies, and the ability to compartmentalize different project aspects by understanding the different project contexts.*

Keywords: curriculum, web development, web technologies, web programming

1 Introduction

1.1 Background

Web technologies and languages have evolved significantly over the past several years. From CGI/Perl programs to HTML/Javascript to PHP/ASP, the web application environment became a specific niche within the computing field. Programming Environments such as ColdFusion automate much of the code work but still require knowledge of HTML and other foundational expertise.

The purpose of this study was to determine optimal persistent course content and teaching methodologies for a university level web development course offered within an Information Technology degree program. The specific research questions were 1) "What elements of a web development course should remain the same over time?", and 2) "What is the optimal course format for effective delivery?" These questions were answered by considering data from previous web development courses, surveying related literature, and interviewing students who have completed the course.

1.2 Overview

In this report we first summarize and explain the rationale for changes in the course delivery. Next, we provide a short survey of what those teaching web development and web technologies courses have done regarding content and teaching methods. Then we present what former students of the course have provided to help optimize the course. Finally, we make some conclusions in an attempt to answer the research questions and provide a vision of what enduring principles, content and methodologies of an effective web development course.

2 Rationale for Course Changes

Table 1 below shows the course evolution over a six-year period. Now, based on the research questions, what changes, if any, should be made to the existing content and methodology?

2.1 Initial Course (2007)

When the IT Program began in 2007, the web development component consisted of a basic course in how to use HTML. The concept of a markup language was new to both the students and the instructors. A markup scripting language is neither learned nor taught in the same fashion as a traditional instructional language such as C++ or Java. Soon, however, it was realized that more would be required; that server-side languages and database connectivity were necessary for developing interactive applications at a level that industry would expect.

Table 1. Evolution of the web development course

(2007) First semester course: Specific HTML tags, CSS commands and some JavaScript
(2009) First semester course: XHTML and CSS conceptual understanding followed by projects involving JavaScript and ending with a basic MySQL/PHP project
(2011) Third semester course: First half of the semester on XHTML/CSS, JavaScript and web server setup; second half of the semester incorporates MySQL/PHP as back-end of HTML/JavaScript projects. About half of the class time, on average, spent working on projects allowing students to ask questions
(2013) Third semester course: Overview of HTML5/CSS3, JavaScript and web server setup followed by intro to XML, Ajax, JQuery before adding MySQL/PHP to the projects. About half of the class period spent as lab to reinforce a specific topic, such as Forms, Events, Draggable, or mysqli PHP class, to ensure students have basic understanding of concept before being required to incorporate it into the programming assignment or project.

2.2 Second Iteration (2009)

The second iteration included adding a server-side component to the course, offering an introduction to PHP and database connectivity via the mysqli class. This added complexity and sheer course content necessitated a reduction in the amount of time allotted for basic HTML/CSS/Javascript. Projects became more involved and all but the best students became more overwhelmed.

2.3 Third Iteration (2011)

After adding the server-side elements to the course content, the course became so complex that it no longer could be offered as an introductory course. Students and instructors alike recognized the need for more preparation. The course number was changed from a 100-level course to a 200-level course and prerequisites included the introductory Python scripting sequence. This also afforded the maturity to handle the plethora of technologies and languages found in the course content.

Concurrently with the content evolution was the change in methodology and assignment structure. Assignments changed from being short and prescriptive to more involved and open. That is, originally, assignments required students to use, for example, the <title> element in a single-page program. During later semesters, instructors would provide a conceptual explanation of a tagged language, provide an HTML language reference, such as w3schools.com or w3.org (the official site), and show how to use them to independently learn specific elements. Students then are able to create multi-page applications immediately following only general assignment guidelines.

2.4 Fourth Iteration (2013)

As before, this "freedom with responsibility" was welcomed by the more prepared students and not well-accepted by those less prepared. Yet this methodology was still considered more optimum than the previous one. To help meet the needs of the struggling students, instructors reintroduced more open lab time within the class period. Assignments and readings were due before class that would prepared students for the lab that would be used to quickly understand the concept during the class period so the longer project assignment would not seem so overwhelming.

In the last five years or so it has become popular to "flip" the classroom, that is, post videos of the lecture on the course site and use the class time for asking questions and working on projects [4]. Many blogs and newsy articles have been written on the topic by not a lot of clear academic evidence as to its specific results. While this iteration of the web development course did not completely adopt class inverting, or flipping, many of the principles of that technique were being incorporated.

3 Survey of Web Development Courses

From the literature, two areas emerged from which to evaluate the course in question: course content and course methodologies and labs. In general, however, there is a noticeable lack of research publicized regarding web development curricula. Connolly [2] noted that only about 1.5 percent of the papers presented at SIGCSE and ITiCSE in the last decade we concerning the teaching of web topics. Considering the importance industry is placing on web programming expertise, this near void indicates an area where research is sorely needed. Our current investigation found most studies to be dated 2004 or earlier.

3.1 Course Content

The literature showed that the evolution of web development, or web technologies, courses wasover time was similar to the evolution of the course under investigation. As web technologies courses matured, client-side languages such as HTML, CSS, and JavaScript had to make room for the server-side components as well. Effective, web-based applications were shown to be those that were database-driven and harnessed the power of the servers via languages and services such as PHP and MySQL. Presented in [10] is a sample course outline that incorporated these web technology elements.

The same professor, eight years later [11] reported how his curriculum had expanded to include an additional course to further apply the web technologies. Ajax, JSON, JQuery and other technologies have emerged only since about 2006 [2], but are prevalently used in industry. It is important that these are included in web development courses for two reasons: 1) industry needs graduates with this expertise and, 2) the students have greater self-efficacy when they know they are learning the current trends in their chosen field.

Noted often in the literature was the observation that the specific technologies, languages, or tools were always going to change and that new approaches would always be emerging. "[T]eaching the underlying concepts conveyed by the language—not the language itself — is what's relevant " (p. 63) [5]. This idea of teaching how to learn the new technologies rather than teaching the technologies themselves was a recurring theme.

Once nice thing about most of the web technologies is, in general, there are options for free and open source programming standards, protocols, languages, and services. For example the entire LAMP stack (Linux, Apache, MySQL, and PHP) is free and open source [8]. This allows easy, unfettered access for students and instructors alike. A few exceptions to this model are the Microsoft products and some of the commercial web page builders. But many of those have free substitutes or "community" versions of their products.

3.2 Course Methodologies and Labs

This subsection discusses what teaching philosophies have worked well and how successful labs are structured. In one study [7], the format of the course was iterative in that the big picture was introduced first project. Then during the second and third projects the concepts were repeated at a more detailed level.

It was recognized early on (1998) that "traditional tools and techniques that organizations have been able to employ with relative success the past for few decades are not applicable for developing web applications" [6], p. 107. One approach [13] was to offer, in essence, a web special topics course titled Internet Application Development (IAD). Rather than a web programming course per se, IAD was an opportunity to gain exposure to various new technologies as they became available.

Another approach to deal with the rapidly changing technologies was to offer the web development course online [12]. Essentially, the same content was covered but with an increased focus on projects over concepts. Project-based learning (PBL) works well online or in a face-to-face mode.

Overall, the consensus was that a top-down approach to teaching this topic is better. One study [9] showed that the time to gain the same proficiency was about half when using the top-down approach to learning. So, taking HTML as an example, students would learn twice as fast when the big picture of a web programming application was presented first before assigning the details of a specific element such as <title>.

4 Student Perspectives

After obtaining university permission through the human subjects review board (HRB), several students were interviewed who took the course in a semester ranging from Fall 2011 to Fall 2013. These data are summarized in the table below by providing representative responses to the items contained in the survey instrument. The intent of the survey was twofold: 1) determine the utility of the course in terms of known job applicability or at least in self-efficacy and 2) learn specific ways in which the course could be improved. (Normal end-of-course critiques were also a part of the study but not included in this report.)

Table 2. Summary of Student Interviews

Interview Question	Response Summary
How will you use knowledge gained from this course?	Since being employed I have used more information I learned in this course than I believed I would; I fell in love with it and now plan on doing it for my career; I've already used it to do freelance web dev
What are the main aspects of the course?	Learning to work with many languages at once, working with databases, and maintaining servers
What aspects were the most useful as a web developer?	Exposure to the different languages and applications, connecting to a database, learning the latest technologies
What aspects were most useful for general technical or other knowledge?	HTML and Javascript will be most useful; it showed how different languages can all come together to make a web site; learning to work with databases and troubleshooting
What could be eliminated without adversely affecting the usefulness of the course?	Reduce complexity of programs and focus on technologies and their integration. [Usually, the answer was "none".]
What, if anything more should be included in the course that is not a part of it?	We learned more than what is expected from most entry level web dev courses; keep introducing the new technologies;
What are your thoughts on a textbook (paper or electronic) compared to using only Internet resources?	About half said a text was needed or beneficial, other half said they didn't need a text because Internet resources were sufficient

5 Summary and Conclusions

5.1 Summary

In this report we presented the evolution of a web development course offered as part of a Bachelor of Science Degree in Information Technology from 2007 to 2013. We then compared existing literature to see what similarities existed and where there might be areas of improvement. Finally, we presented what students have thought about the course over a period of the last three years the course was offered.

5.2 Conclusions

After researching web development curricula, seeing the relatively few published papers on the topic, and personally teaching the course, we can say that one thing is as clear now as it was in the early days of teaching web development. "[T]he only thing that seems clear is that web development and its associated employment options may be changing too rapidly to capture in a university curriculum" [3] (p.5).

This was a repetitive theme: "Teaching web application development is not easy. In the end, students must be competent in a variety of seemingly disparate topics, as well as be able to put them all together into a single application" [1] (p. 97).

To add to the complexity, student preparation will vary depending on whether the course is being offered to Information Technology majors or Computer Science majors. Just because a set of prerequisites has been met does not guarantee equal preparedness. One course for both majors will complicate matters more.

That being said, there are several tips we can apply regarding the teaching of web development courses.

- It is more important that the students learn how to find the solution than to know the solution [7]. Learning how to learn should be one of the primary goals of a web development course. The multitude of technologies required to develop and implement a web program offer a great opportunity to foster systems thinking and instill effective troubleshooting techniques.
- Use the top-down approach. Rather than teaching specific language constructs, teach the concepts. Even at the project level, assign a technology-inclusive programming application and make the follow-on project over the same technologies but require a more refined version.
- Assign short labs to be completed in class that will provide a basis for independent work for the more complicated integration of the technologies.
- Do not afford students the opportunity to get behind in their work and learning. In this course, allowing late assignments has proven to set them up for failure. They may not like it at the time, but when the semester is over they will definitely appreciate you for it.
- Use open source technologies whenever possible to prevent licensing issues and allow open availability to working anywhere.
- Include in the course all aspects of the web application—both server-side and client-side elements, web server concepts and applications (e.g., apache configuration, php.ini configuration), database connectivity, etc. It is better to have students learn the overall picture of how the various technologies work together, that to just know how to program in a single language.
- Design the course so that new technologies can be incorporated as they emerge (e.g., include coverage of JSON, JQuery, Ajax, HTML5/CSS3, etc.)

Designing and teaching a web development course is always going to be complex because of the myriad of essential components required to make an effective web application. But we are up for that task.

6 References

[1] Adams, D. R. (October 2007). “Integration Early: A New Approach To Teaching Web Application Development,” ***Journal of Computing Sciences in Colleges(23:1)***, pp. 97-104.

[2] Connolly, R. (May 2012). ”Criticizing and Modernizing Computing Curriculum: The Case of the Web and the Social Issues Courses,” ***Proceedings of the Seventeenth Western Canadian Conference on Computing Education,*** pp. 52-56. http://dl.acm.org/citation.cfm?id=2247587

[3] Courte, J. E. (October 2004). “The Difficulties of Incorporating Web Development into a University Curriculum,” ***The Dream and Reality WWW @ 10 Conference,*** pp. 1-5. http://www10.cs.rose-hulman.edu/Papers/Courte.pdf

[4] Craven, P (November 2012). “Flipped Classroom Experiment” http://simpson.edu/2012/11/flipped-classroom/

[5] Kafai, Y. B. & Burke, Q. (September 2013). “Computer Programming Goes Back to School,” ***Education Week,*** pp. 61-65. http://www.edweek.org/ew/articles/2013/09/01/kappan_kafai.html

[6] Lim , B. B. L. (July 1998). “Teaching Web Development I Technologies in CS/IS Curricula, ” ***Proceedings of the Twenty-ninth Computer Science Education Technical Symposium,*** pp.107-111.

[7] Liu, J. & Qian, K. (June 2006). “An Incremental, Spiral, Topic-Driven Approach: Designing labs and projects for a Web technology course,” ***Proceedings of the 2006 International Conference on Frontiers in Education: Computer Science & Computer Engineering.*** pp.67-74.

[8] Mudigonda, S. & Buerck , J. (May 2012). "The Design Of An Undergraduate Web Application Development Course Using Free Technologies And Learning Materials," ***Journal of Computing Sciences in Colleges(27:5)***, pp. 131-139.

[9] Saito , D. & Yamauara , T. (December 2013). "A New Approach to Programming Language Education for Beginners with Top-Down Learning," ***International Journal of Engineering Pedagogy(3:S4)***, pp. 16-21. http://dx.doi.org/10.3991/ijep.v3iS4.3216

[10] Wang, X. (October 2006). "A Practical Way To Teach Web Programming In Computer Science," ***Journal of Computing Sciences in Colleges(22:1)***, pp. 211-220.

[11] Wang, X. (April 2014). "Design, develop and teach the second web programming course in computer science curriculum," ***Journal of Computing Sciences in Colleges(29:4),*** pp. 52-59.

[12] Wang, Y. D. (October 2011). "Teaching Web Development at a Distance," ***Proceedings of the 2011 conference on Information technology education,*** pp. 91-96.

[13] Yue, K. & Ding, W. (June 2004). "Design and Evolution of An Undergraduate Course on Web Application Development," ***Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education***, pp. 22-26.

Development of Prototype System for the Art Appreciation Support System to Integrate Visitors into Art Objects in Real Time

Koji Fujita, Takayuki Fujimoto
Graduate School of Engineering, Toyo University
Kujirai2100,Kawagoe-City,Saitama, Japan
s46d01300026@toyo.jp, me@fujiotokyo.com

Abstract - *Recently, many art museums has support system for appreciate. This is provided to visitor to understand many art objects. Those systems are used sound or video. However, those systems are many problems. Those systems don't lead to understand of the art objects for visitor. In addition, those systems aren't easy use for visitor. From of reason, this paper developed interactive system that every visitor can use very easy and can understand of the art objects.*

Keywords: chroma key processing, art museum; scale processing;

1 Introduction

Recently, there is a museum in many places. The museum can see a historic document and many display work. The art museum is one in the museum. The art museum has many places because there is the genre of much art works. The art museum exists from small atelier to the big art museum. From of this, the art museum is expected that many visitors come to art museum. We show the result of research by "Lifemedia Co., Ltd".[1] The person who goes the art museum a year is 32%. For the result, we understand that many visitors don't go to the art museum. We thought about the cause. Visitor comes to art museum to look at the art objects. Generally, the view method only looks in front of an art object in the art museum. It is some reasons. For example, visitor can't touch the art object and can't take the art object because they are so precious. Art appreciation is the one of the entertainment. However, visitor can only look passively the art objects. Other entertainment is nothing in art museum. Based on these points, in this research, we developed art appreciation support system that it is high entertainment and like body movement.

2 Purpose

Recently, there are many support systems in art museum. For example, voice guidance system. This is guidance system to put in headphone and it guides about many art objects. Many art objects don't explain easy. Thus, this support system is provided for visitor to understand art object more and look at art object more. Another support system is portable information terminal. This system is rented in art museum. It is the system that operated by hand and it support to look at art object. However, those systems have been used the visitor who is interested in an art object. For example, when a visitor who is not interested in specific art object came to the art museum, the visitor look only at the art objects with the interest. The reason is simple thing. The visitor doesn't look at the art objects which is not interest in the art museum. Because of this, the visitor doesn't look at art object with attention. In some case, when the visitor is not interest, It is assumed that the visitor spend boring time. We try to look at general information of display work in the museum. It is put on the information of painting below. (For example, title of painting, the name of the author.) Many art museums explain only of this level. In addition, it is rarely written easy the history background of art object and background of the creation. However, many art objects aren't explained moreover. For the reason, the visitor who doesn't have knowledge about the art object don't lead to understand of display of work. Therefore, the action finishes to look only passive art appreciation. For the result, those problem lead to decrease understood of art object. This is important problem to be greatly lost of the entertainment. In this result, it has the possibilities that lead to decrease interest to art object. From of this, in this research, we developed the system which didn't only do passive art appreciation when the visitor appreciate painting. The visitor doesn't look only at the painting in front of painting. The system can look like for body movement. Moreover, the system integrates real time video of visitor and painting. In addition, we developed support system of art appreciation that can project in real size of painting when the visitor is projected.

3 System summary

We explain the summary of the system in this research. This system imagined the situation that a visitor is watching in front of a painting in the art museum. We imagine system environment that the visitor is projected monitor to put on a painting by side. Moreover, the video of a visitor projected to a monitor to compare the viewer with the painting. The scale of visitor is projected real size in the painting. In this paper, we performed prototype system to set same environment.

3.1 About enviroment of system run time

We assumed run environment in the real art museum. We though environment like figure 1.

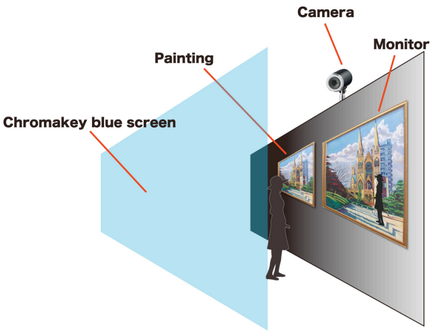


fig. 1. Environment of system

Look at figure 1. Monitor is set by a painting to watch. Next, the monitor is projected the video of scale processing in real time. Therefore, we set camera on a painting to take a visitor. In addition, the system use chroma key screen to integrate the visitor and the painting. We set background of the visitor. The system set the device to push visitor's height to process the scale of the visitor. The system is performed in environment.

3.2 Development of system

We explain environment of development in our research. In this paper, operating system is used "Mac OS 10.8.5". Programming language is used "C++". Library is used "OpenCV2.4.8". In addition, compiler is used "Xcode5.0" of integrated development environment.

3.3 About the run of the system

This time, we construct the environment limitlessly close situation like figure 1. The device which we used is "web camera", "chroma key blue screen", "monitor", "electronic computer".

- web camera(ELECOM UCAM-C0220FB)
- monitor(W32-PD2100)
- electronic computer(Mac OS 10.8.5)
- chroma key blue screen

We think that the visitor is watching the painting in front of a painting. In addition, a visitor and camera has distance of 14.77 feet. Moreover, the chroma - key screen set at 13.12 feet in height, 13.12 feet in width. We show the figure 2, figure 3, to be set system environment.

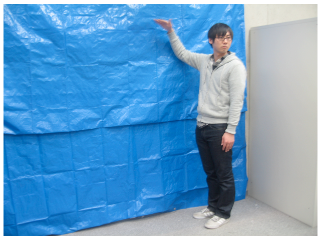


Fig. 2. Chroma key screen to be set

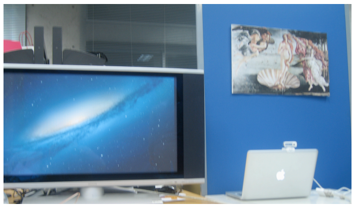


Fig. 3. System environment to be set

The chroma key screen which perform to integrate real time video is set blue in this time. This system can integrate everything color. However, the general video integrating method is used near to a complementary color of the human skin. For the reason, we used this color. In addition, we used this painting. The painting is drew "Sandro Botticelli". The title of painting is "Birth of Venus". The painting is displayed in front of the visitor. Furthermore, the monitor which can look the real time video of the visitor to integrate in the painting is set by the painting. We show the real environment in figure 4.

Fig. 4. Birth of Venus

3.4 System flow

We explain the flow of system. In this system, we used chroma-key processing method. This method is projected to the monitor in real time video. It integrates the painting and the visitor. We show below the step and show the details in figure 5.

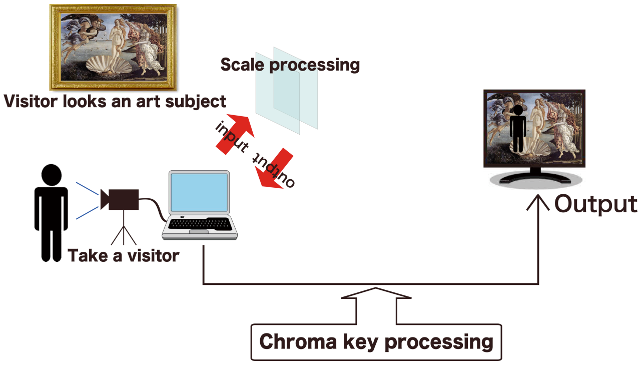


Fig. 5. System flow

・STEP1:

The visitor push your height.

・STEP2:

The camera is taken the visitor who watch the painting.

・STEP3：

The video which was taken is performed the mask processing.

・STEP4:

The mask processing video and a painting of background are integrated in real time.

・STEP5:

The video which was integrated is projected the monitor.

In this system, we used color of blue in background when we perform the mask processing. We set the maximum of hue and the minimum in program of system. In addition, we set the number to integrate background of blue.

3.5 About scale processing method

We explain the method of scale processing in this system. In this system, the method push visitor's height before visitor use this system. In this time, the program is changed picture size for visitor's height. In this system, we assumed height of Venus to be 5.41 feet. If visitor was higher than the Venus, the monitor is projected the background image which we resized and reduced. The background image calculated the ratio. If visitor is smaller than the Venus, the background image is projected the image which was enlarged. This is performed in the program for condition. Moreover, it is performed every 1cm.

3.6 Result

We explain about the result of run. We show the result when author is used this system. The height of author is 5.71 feet. The author is higher than the Venus of painting. We show the result of run in the figure 6.

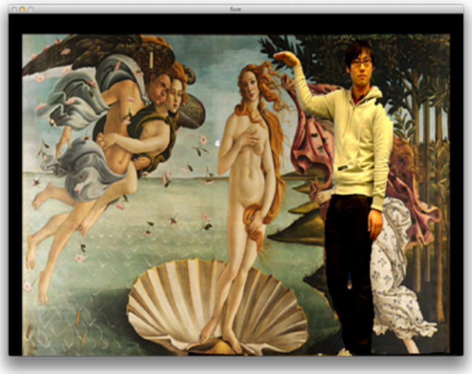


Fig. 6. Result of run

For the result of figure 6, we understand that the Venus is smaller than the author. Moreover, the result is shown that the visitor can perform action of the body movement. In addition, the visitor can performed virtual movement to a person drawn on the painting. For the reason, the visitor cannot be only passive appreciation. Therefore, the visitor can perform body movement-like appreciation.

4 Evaluation

4.1 About a subject experiment and questionnaire contents

In this research, we evaluated the system. Method of evaluation performed to use our system. The subject is 13 people. After the subject finished to use, the subject replied a questionnaire. The contents of questions is this.

[1] About your gender

[2] About your height and age

[3] About use frequency of the art museum

[4] About visibility of the system

[5] About novelty of the system

[6] About future of system

[7] About entertainment of the system.

[8] Did you become interested by appreciation using this system?

[9] Did you become interested in the art museum with this system?

[10] Did you want to go the art museum with using this system?

[11] Which became interested in art viewing by an art museum and this system?

In addition, the ratio of the man and woman are man 7 people, woman 6. In addition, we show the ratio of age and height in figure 7, figure 8.

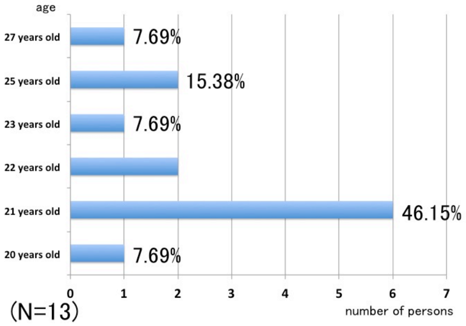


Fig. 7. Ratio of age

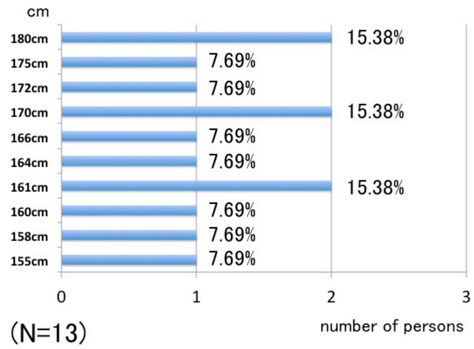


Fig. 8. Ratio of height

4.2 Simple evaluation result of the system

First, we explain the result of simple evaluation, when we performed the subject of an experiment. We asked question about use frequency of the art museum. We show the result into figure 9.

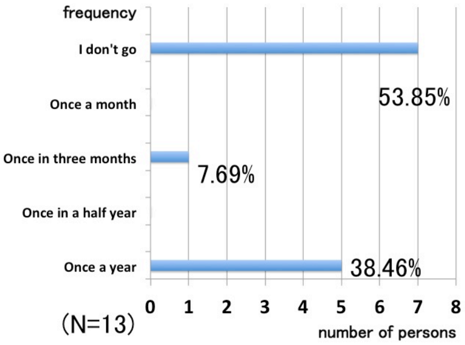


Fig. 9. use frequency of the art museum

From the result of figure 9, the subjects half replied that they never go to the art museum. In addition, the subjects whom they answered, "I went once a year" is about 39%. For the result, we understander that many subjects didn't go to the art museum. Next, we show the result in figure 10 about "visibility of this system".

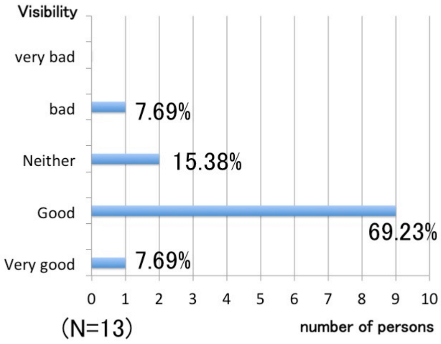


Fig. 10. About visibility of system

From a result of the figure 10, the subjects replied about 77% that this system is Very good" or "Good". Because of this, we understood that this system has good visibility. Next, we show about the novelty of system in figure 11.

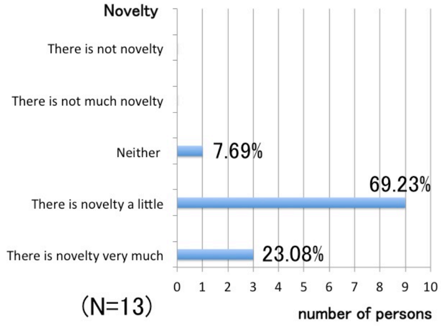


Fig. 11. About novelty of system

From a result of figure 11, the subjects replied about 69% that this system has "There is novelty a little". The second most the answer was "There is novelty very much". This answers were about 23%. For the result, we understood that the subjects of about 90% felt novelty of this system. In addition, we checked other answer. There is only an answer "Neither". Therefore, there is not the person who felt that ”there is not novelty”. For the result, we get answer that this system can feel possibility as new support system, when the visitor watch a painting. We show the result about the future of system in figure 12 and the result about entertainment of system in figure 13.

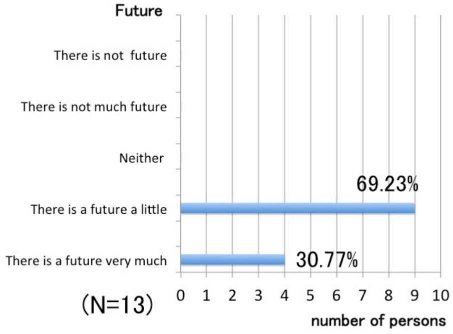


Fig. 12. About future of system

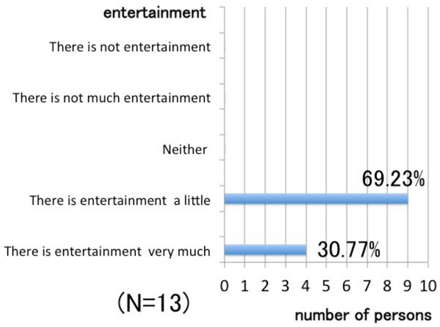


Fig. 13. About entertainment of system

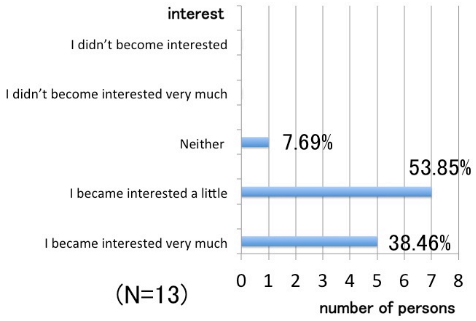


Fig. 14. About interest by appreciation using this system

From the result of figure 14, one subject replied "Neither". Other all subjects replied "I became interested". Accordingly, we understood that this system could become interested for the visitor, when the visitor watched a painting. Thus,this system was able to improve the interest about the painting. This said that we was able to improve passive appreciation. This is possibility to improve recently problem in the art museum.

4.3 About the interest in art museum by the use of this system

We show the result of simple evaluation about interest in art museum when the subjects used this system. First, we asked about "Did you become interested in the art museum with this system?". We show the result in figure 15.

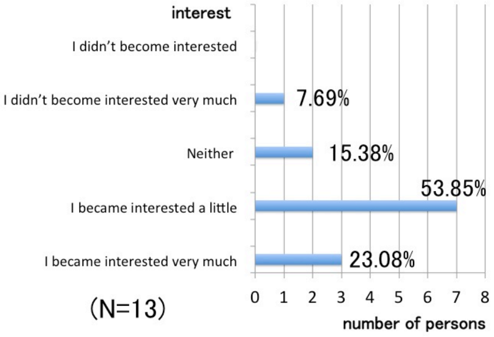


Fig. 15. About interest in art museum to use this system

For the result, the subjects replied seven people,"I became interested a little". It was about 54%. The second most answer was "I became interested very much". It was 3 people and about 23%. From the result, we understood that the subjects of about 77% could feel interest in the art museum to use this system. For the reason, this system can only use to improve the entertainment of system or art of work. In addition, this system is expected as art appreciation support system. Moreover, It is expected the use as the education support system. Next, we asked about "Do you want to go to the art museum where this system was set". We show the result in figure 16.

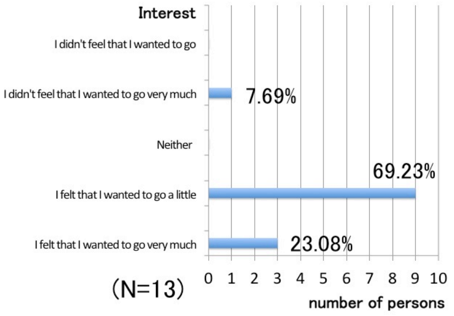


Fig. 16. About interest to the art museum for using this system

From this result, the subjects replied "I want to go very much" or "I want to go a little". The answer is 12 people and about 92%. We understood that this system got desire for the appreciation in the art museum where this system set. In other hand, The subject who felt, "I don't want to go very much" was 1 people and about 8%. This subject said that this system seems to be interesting. However, the subject think it was ashamed when the subject use in the art museum. Therefor, this opinion is new problem in this research. This system use in art museum when the visitor appreciate. Moreover, this system use for body movement. Thus, the subject may feel that this point is ashamed. This problem will want to improve future work.

4.4 About a comparison evaluation with the general art appreciation

We asked that the subject replied about the evaluation to compare general art appreciation and this system, when this system use. We show the result in figure 17.

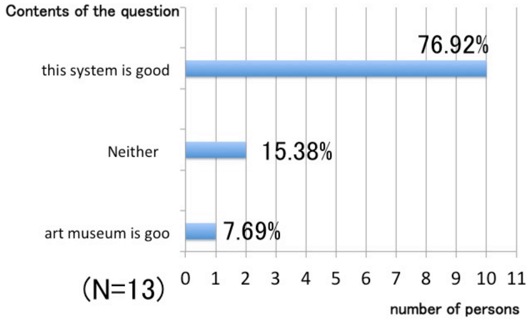


Fig. 17. Comparison of art museum and this system

From this result, the subjects replied 10 people,"This system became interested", when the subject compare general art appreciation and this system. It was 77%. On the other hand, the subject replied only one,"general art museum became interested". It was 8%. For the reason, we understood that this system could became interested than general art appreciation.

4.5 General comment of simple evaluation

The subjects used actually the prototype system which we developed in this time. In the result, we got answer that the subjects replied affirmative answer about all question contents. They were more than half. This result understood that this system is good than general art appreciation. Furthermore, we understood that this system can get interest about the painting and improve entertainment. Here, we give the opinion when subject of a woman use this system. The subject was looked the painting of "The Venus", when she used this system. She said opinion,"Venus has better style than me" and "Face of Venus has smaller than me". This opinion can understand that the visitor compares oneself with Venus to use this system. It cannot feel this in the general art appreciation. For the reason, we understood that the visitor can appreciate the painting to do objective and subjective. On the other hand, men of subjects said opinion like this. We thought that this opinion will be future work. The subjects said that this system has hight entertainment. However, there is not the guide system which is direct about a painting. Therefore, the subject said that it was difficult to understand the painting. About this point will want to improve as future work. In the future, we will want to add voice guidance system and text guidance system.

5 Future work

We explain about future work. About one problem is scale processing method. This system require to make more smaller. At this stage, we set the standard value at the length of a particular part in the painting. This system is performed scale processing to compare the painting with the visitor. The current system can process by a 0.39 inches unit. However, the range is less than 5.74 feet more than 5.09 feet. From of this, other visitors is projected a constant background image. In the result, for visitor of short can not perform close scale processing. In the future, we will want to develop the system that more careful scale processing is possible. Second problem is system structure. The system structure can only use one visitor. For the result, the use in plural people is not possible by this system. For the result, this system is performed scale processing only one visitor because we depend on the specific height of the one. In the future, we will improve to the system that can use in plural people. Finally, in this system, we put an important point to improve entertainment characteristics. In this future, we will improve to understand display work about art museum more. In addition, we will want to perform system construction.

6 Conclusions

In this paper, we performed to develop the system that the visitor and painting integrate real time and evaluation. Moreover, we proposed new method to look in the art museum. The current art appreciation was finished to look only painting. For the result, we could not feel the painting for body movement. However, we can perform to use in this system. Moreover, we can feel high entertainment in art museum. In the future of this research, a visitor only enjoys through the monitor of the place. Therefore, we will want to improve good system more. In addition, we want to show the sound guidance and text guidance about the painting when the visitor is watching the monitor. Moreover, the use of application system which use the smartphone is necessary. Therefore, the future of system can take the image or video in the visitor's smartphone when the visitor uses this system. In addition, when the system finished, the visitor can get the image or video in your smartphone. Finally, we will think that we want to develop the system as art appreciation support system more.

7 Related research

Recently, the research about appreciation support system in the art museum exist many researchs. For example, the appreciation support system which can use with your cellphone exist many more. In the paper of Hirasawa[2], the visitor use system which can use with personal digital assistant in the museum and the system perform the support about the display of work. This system can perform explanation of art work (For example, show the map, voice guidance, Visual guidance). However, support system of personal digital assistant must carry visitor oneself. Furthermore, The completeness of interface is important when the system build as an application system in a device. The elderly people who are inexperienced in a device in particular is expected that the user don't know operation how to use. For the result, the system doesn't make sense. On the other hand, this system watches only the painting in front of display of work. The special personal digital assistant doesn't carry. Because of this, our research can say that there is superiority than those researches. On the other hand, a research which does not use special personal digital assistant is performed. For example, we show the research that Yamashita work[3]. This support system set 400 transmitter which had infrared LED built-in. The transmitter is set In about 44.4 square feet. For the system, the user is measured your position. Our research is projected to the display and the special device is not necessary. The user can play the game to use this system. This system is good that various information is received to use positional information of the user. However, there is the problem. Organizer must set a lot of infrared sensors. In addition, The cost is very high and it is not easy to set system environment. On the other hand, the special device is not necessary in our system and our system is easy to set system environment. For this reason, our system is superiority to compere than those system. Moreover, the system which support to watch the painting after having improved entertainment do not exist with the thing which an author knows. In the result, we can say that our system is superiority and novelty.

8 References

[1] Lifemedia, Inc, “Survey about the art museum”, http://research.lifemedia.jp/2013/11/131113_art.html

[2] HIRASAWA Yasufumi, MATSUKAWA Takashi, KAWATA Takao, KOMINAMI Masanobu, "Development and Evaluation of iPad Guide System for Museum Exhibits", 2012 Japan Society for Educational Technology 36, pp.89-92, December.2012.

[3] Yamashita Jun, Hiyama Atsushi, Nishioka Teiichi, Kuzuoka Hideaki, Hirose Michitaka, "HK-2-2 Ubiquitous Gaming : A Study of Ubiquitous Information Supporting System in the Museum", Proceedings of the IEICE General Conference 2005 the basics, border, SS-12-SS-13, March.2005.

[4] Kusunoki Fusako, Yatani Kouji, Ishikawa Aoi, Ishiyama Ayako, Yamaguti Takako, Sugimoto Masanori, "PIBOOK : COLLABORATIVE LEARNING TOOL AT A MUSEUM", Technical report of IEICE. HIP, 103(166), pp. 11-15, June.2003.

[5] YANAI Masahiro, TACHIBANA Shoju, SHIBA Masahito, OKADA Yoshihiro, "An exhibition support system using gaze direction measurement and handheld device position tracking", IPSJ SIG Notes. CVIM, 2008(3), pp. 361-364, Jan.2008.

[6] SATOH ICHIRO, "Experiences with the M-Spaces Location Model in Context-aware Services for Museums", IPSJ Journal, 49(2), pp.797-807, Feb.2008.

[7] Ayako Ishiyama, Fusako Kusunoki, Shigenori Inagaki, "The evaluation of museum support content and user interface", IPSJ SIG technical reports, 2013-HCI-153(5) , pp. 1-5, May.2013.

[8] OKUMOTO Motoko, KATO Hiroshi, "The Learning System Linking Pre-Visit and Museum Learning Materials", Japan Society for Educational Technology, 36(1), pp. 1-8, July.2012.

SESSION

RECRUITMENT, RETENTION AND MENTORING + ENROLLMENT STUDIES AND MANAGEMENT + TOOLS AND CASE STUDIES

Chair(s)

TBA

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Increasing Diversity in Engineering, IT, and Computer Science

M. Barjaktarovic
Department of Computer Science, Hawai’i Pacific University, Honolulu, Hawai’i, U.S.A.

Abstract – *There is an interest in increasing diversity in traditionally very homogeneous and rather closed professions of engineering, IT, and computer science. The National Science Foundation (NSF), government, corporations, as well as institutions of higher learning all have this issue on their agenda, for various and differing reasons. Although many efforts have been made over the years, these professions are still not diverse enough and numbers of some minorities have even decreased. In this paper we discuss some of the potential barriers to diversity and our approach to overcome them as a female faculty member in a computer science department. Strategies, pitfalls, and successes are presented.*

Keywords: diversity, minority, engineering, IT, computer, science

1 Introduction

There is much discussion about numbers of technical labor force in the U.S. Some organizations, such as IEEE, claim there is no shortage [14]. Some other organizations and industry claim there is a shortage [15]; for example, Senior Vice-President of IBM Technology group advocates active measures to increase the numbers [19]. Industry in general claims there are not enough “good graduates.” “In a survey of 2,001 students or recent graduates (18- to 24-year-olds) and 1,000 hiring managers, fewer than two in five hiring managers who had interviewed recent graduates in the past two years found them prepared for a job in their field of study” [20]. Perhaps this data indicates that we do graduate enough students, but with insufficient skills for work in industry.

These questions are very important to consider when attempting to assure a country’s economic prosperity and security, for example providing workforce capable of competing in the global market and digitally protecting and defending the country's resources [1] [2]. The U.S. government itself acknowledges a shortage of technical workers, especially in the area of cybersecurity; expresses concern about it; and supports efforts to increase numbers of graduates. Effort to increase understanding of effective undergraduate Science, Technology, Engineering and Mathematics (STEM) education produced analysis of effective practices, directions for future research and suitable evaluation criteria [3] [4].

The consequence of these concerns and efforts is considering populations that traditionally did not supply majority of technical workers: women, minorities, older workers, veterans, and workers with disabilities. Tapping into these resources can increase the numbers as well as the quality of creative input, as these relative “newcomers” change the nature of the field. However, many approaches have been tried previously in engineering-like fields and those fields are still not very diverse. Some numbers are declining, for example the number of bachelor degrees in computer science awarded to women [29]. It is still not clear as to how to achieve the goals. National Science Foundation (NSF) held Engineering Ideas Lab in March 2014, to brainstorm and address these issues [1].

Facing diversity might be necessary in situations where formerly underrepresented groups become the majority. Some states have predominantly minority populations. For example, in Texas in 2012, “more than 55% of population was of race other than non-Hispanic white” [18].

In states where diversity is not as pronounced, facing diversity may not be an urgent topic at many institutions of higher learning. However, learning institutions are typically interested in increasing numbers of students and thus increasing tuition revenue, which can lead to a willingness to tap into new markets. In addition, relatively recent availability of grant money for improving STEM education is a motivator for higher-level institutions to put effort towards such goals. Currently, there are no incentives for faculty to spend any time on valuing and increasing diversity in their classes. One important research issue is to devise structural changes and administrative incentives, e.g. tenure and promotion criteria, that values diversity [25].

Even if underrepresented groups are enticed to join STEM fields, the question remains how to keep them engaged and graduating successfully. What are the barriers to entering the field, retention, and professional success?

In this paper we outline several measures that we use in our own classrooms to mitigate these issues. We have experience teaching in Hawai’i for the last 10 years in public and private universities and community colleges, to students of varying ages and socio-economic backgrounds, in a very diverse population including many Native people.

2 Generation I

One of the prime targets for recruiting into STEM and thus increasing numbers and diversity is the younger generation. There are several terms for people who have never known a world without computers and cell phones. Common terms are “Millennials,” “Generation I,” or “Generation Z,” grouping all people born in the Age of Information, approximately starting in 1980s and late 1990s, respectively. “I” stands for “Information Age” and “Internet,” which is the daily staple of Generation I; “me” and “one” as indicated by spending a lot of time on computers and less in social interactions [12].

New generations in the U.S. have certain preferences about studying. They do not feel strongly about studying STEM fields. Nationally, when high students are asked why they do not want to study STEM fields, they complain that such fields are “boring” [5]; that students often do not know adults who work in STEM field and if they do meet any, such adults seem “one-dimensional nerds,” and that “students do not know how to work in teams” [27]. The picture is even worse for minorities such as women and students from various underrepresented backgrounds [7]. Another barrier is lack of mathematical and scientific training in American K-12 education; the U.S. ranks low in mathematics and reading, ranking 23-29 out of 34 developed countries, as summarized in [8] using data of the PISA global assessment of students [9] and the latest PISA 2012 assessment [10].

Traditionally, much of STEM education was predominantly lecture-based, but students today do not like lectures. Students are enthusiastic about hands-on classes and activities and working in “fun” teams. Active learning classes, discovery-based laboratories and course-based research provide personal experience that makes theoretical concepts clearer as well as skills for employment and future education [2]. Students also need to feel as a part of the STEM community and capable of working in teams. These practices are core components in traditional Native education. Hawai’i is a highly diverse area where many different groups successfully work together, and it has much to offer in terms of success in working together.

Underrepresented groups have their own particular needs. For example, the unique need of female workers is pregnancy leave. All underrepresented groups need visible and socially endorsed role models. For example, President Obama is a great inspiration for many African Americans, many children now think it is common to see an African American in a position of power.

3 Creating Community; Networking

Students are more likely to succeed if they feel a sense of belonging to their “community.” The community provides the practical support and social niche necessary to withstand the demands of academic life. The community for college students is their peers in classroom, as well as their instructors. During the very first class of the semester, we spend about 10-15 minutes, depending on class size, introducing each other – our name, major/year, our goals. Students are then asked to spend five minutes and introduce themselves to one person they did not know before class. In reality, this step takes about ten minutes, until the enthusiastic chatter starts calming down. If a student came to class with “friends” and is sitting next to them, they will have to get up and move to another location in the room to meet new people. Very often, we have a “quiz on names”. In some classes, we ask students to introduce each other. This very simple gesture leads to more comfort in the classroom and more collaboration, inside and outside of classroom. We do many activities in teams during class. We also tell the students to choose their study partners and to study in groups. This initial introduction helps break the ice and bond the students.

Another fine point in “belonging” is sense of importance to the community, of oneself and others; and sense of tangible and practical need to create and maintain a functioning community. After the students finish introducing themselves, we ask: “Why is it important that students know each other?” Sometimes students answer: “Because we are competition to each other.” Very seldom a student will answer that students are resources for each other. Such entrepreneurial way of looking into one’s peers typically has to be introduced. We tell the students that they are business connections to each other from now on; that “many good things” can result from peer collaboration. Everyone knows that jobs, projects and many other resources come from human networks, however, younger students from middle- and low-income families typically need to be reminded that this model holds valid for students as well. We suspect that students do not think of themselves as “grownups” and “business people” and need to be encouraged to build that self-image. We also tell students that their success and networking is important to the institution, since we will be tapping into their success via donation drives, placement of graduates, alumni relations, and other venues.

Another message given to students is that they should observe each other in class and be aware who they want to work with and who they do not want to work with. This measure ensures that all students are on their best behavior and trying their best to succeed in class. This is important to students as they will be doing many projects in teams, and they pick their teammates. Belonging to a team greatly increases retention.

The instructor helps to maintain this sense of belonging by remembering students’ names and details about their lives and academic progress and interests. For example, a student with potential problems at home or work (for example, sick family members, plumbing issues, deadlines at work) is allowed to have their cell phone ring during class (as long as they take it and go outside of the room). A student with particular interests can be alerted when something relevant is happening in that field. In general, all students are alerted to and discuss in class events of major importance for the field, such as Heartbleed virus outbreak.

Another venue for increasing the bond of the community is offered via various campus organizations and clubs, as well as other venues for professional networking. Students are encouraged to establish technical or social clubs on campus as well as to join various professional organizations. One incentive for joining is the prospect of employment. Another benefit is a more interesting social life, higher learning, and more collaboration with classmates and instructors. Working together through trials of programming competitions or organizing events bonds students. As the long-term adviser for the Computer Club, we found it challenging and rewarding to stand by and let the students govern themselves, learning in the process about diversity, team work and appreciating underrepresented groups. The last two presidents of the club were female, and the club was more active than before.

4 Collaboration and Co-Creation

Traditionally, engineering education started in military schools. U.S. Army Engineer School was the first engineering program in the U.S. [16]. In 1898, Norwich University was designated, "The Military College of the State of Vermont" by the Vermont Legislature, and in 1907 Electrical Engineering Department was established [17]. The historic legacy of these facts is that approach used in engineering education is similar to that used for soldiers: obedience, control over body and mind, and top-down approach. There is no sense of equality and collaboration between different “layers” in this hierarchical model. Many Native populations, such as African, as well as women, prefer to work in rather egalitarian groups and co-create together. Team work involves co-creation, but not necessarily in egalitarian terms. Some societies, such as many Asian societies, are experts in team work but not egalitarian co-creation.

An egalitarian co-creative process involving team work does not necessarily proceed naturally in the U.S. and often has to be worked on with perseverance. First, the U.S. culture is geared towards individualism; second, it values authority yet on the other hand finds pleasure in teenage rebelling against authority, thus propagating “underdog” versus “upperdog” mentality and preventing increased productivity.

We observe that students like to view themselves as important, as is common for Generation I and in entitled environments. It is important to honor that. We give students choices: when is the homework due, what they would like to work on, what their project will be. We believe that it is very important to teach them to govern themselves because it leads to greater success in technical fields and opens the door to management and business careers. Technical work is usually done in teams and requires communication, negotiation, and self-governance skills. Excellence in these skills is required both for professional and social success, since they are considered socially desirable opposite of “nerdy” behavior and a coveted sign of leadership and business success.

Increased self-governance is intended to increase self-responsibility, which may not be clear to students until they envision the end product of this approach. The balance has to be maintained: although students have a voice, and a very important voice, they have less experience than the instructor and should respect and listen to the instructor. This can be a challenging task in the environments where students feel entitled to form self-governance in the direction of increased lack of responsibility. This behavior indicates that students do not have clear ideas about their professional future. The instructor’s challenge is to stay in charge of the class and keep the class focused during co-creative process, for example by repeatedly presenting the grand picture and motivating with examples of real-life benefits. Students themselves can be excellent sources of affirmative evidence encountered in their interviews, internships, and jobs. (It is possible that a student has more experience than the instructor, if the student is an adult learner who has been in the field for many years and is in class simply “to get a diploma;” in which case the student assumes a more active role in the classroom.)

In our experience, students are used to top-down approach, which is based on a subtle fear of the instructor as “someone who can hurt you,” e.g. give a lower grade or publicly shame you. In our society, “power” seems to be measured by how much we can hurt others. However, this approach may not lead to greatest productivity and creativity. The alternative of peaceful cooperation based on accomplishing common goals (e.g. learning class material) seems more appropriate, efficient and productive. Working on commonly agreed goals increases self-motivation and thus speeds up and eases the process.

The fine line between collaboration and top-down approach can be very difficult to maintain, as students typically simply do not have experience in shared governance. In our experience, students themselves can be invested in the top-down model. If the instructor is not “on the top," the students may try to take that role. If the administration supports the students in anything they do, the instructor has to be resourceful to avoid repercussions yet teach students about increased self-responsibility. The other end of this problem is seen in large institutions where students are simply “a number” left to the mercy of the instructor. On this end of the spectrum, students may need to be taught to represent themselves with confidence. In any case, students are taught the lesson that “great power assumes great responsibility.”

5 Incentives and Rewards

Traditionally, STEM education is quite rigid in prescribed learning materials. For example, every computer science major has to take calculus, although they may never use it again. Native educational practices are very different. In Native teaching, students choose what subject they want to study, and are taught only if they express interest and dedication. The subsequent training is very hands-on, real-life.

It is my belief that the self-selective and quickly meaningful Native approach can lead to better results with all students, and especially underrepresented populations. The first step is to motivate the students. Students are typically not aware of careers in STEM and their paying potential. Introducing videos, images, slides, and guest speakers makes classes more relevant, interesting, and inspiring. Students do appreciate prospects of higher income and "interesting" careers and typically self-select to work harder to achieve it. They are motivated by tangible goals and real-life examples.

Choosing the right rewards is very important, as rewards are highly motivational. In traditional STEM education, rewards came as "passing the test" which often consisted of memorizing facts. Memorization is considered boring and not creative. Today, problem solving and creativity are expected. In Native educational practices, rewards also required passing some "tests." For example, Native American children in the Plains tribes were quizzed on their bow and arrow prowess by having their food hanging high in a tree on a thin string. In order to eat, the child had to be able to shoot the arrow precisely to severe the string. Current U.S. generations, used to words of praise even if they are failing, would find such tests too harsh and thus require new reward strategies.

Native learning often involves lessons designed to produce failure, as failure often brings most insights and learning. Modern U.S. generations tend to feel very badly about "failing" and want to "succeed" and be rewarded at every step, and even at the very first steps. It requires some training to teach skills of self-introspection and the ability to backtrack and rely on basic concepts; to enjoy the learning process and not take it personally as signs of being "less than," which all students can experience and especially underrepresented populations are more prone to. In a paradoxical way, teaching by using "faulty" examples can put all students on equal footing and bond them stronger. we suspect that it is so because it is highly likely that no students have any experience with this approach, and also because it is presented as a "game." For example, giving students a buggy program and asking them to trace it and fix it usually makes students feel empowered.

One of possible rewards is public recognition, belonging to a community, and sense of accomplishment. Since Generation I is about "I," we find it necessary to include some praise. For example, it is very effective to showcase all student homework, but anonymously, and mention good points and bad points about each piece of work, so that nobody feels "lesser" and good students are encouraged. we practice this strategy in beginner programming classes, showing students' code. This illustrates many possible solutions as well and teaches useful technical lessons.

We also like to use the Socratic approach to teach and ask questions in class. It is advisable to ask the class beforehand if they consent to answering questions one after another. Shy students can usually be coaxed to participate with some gentle probing. Some students may prefer to remain silent. Often an older male can be intimidated by a female faculty and choose to save his face by remaining silent. It is important to acknowledge everyone and find ways to entice them to participate in some way.

6 Written, Spoken and Visual Messages

There is certain terminology in STEM fields that undermines diversity. Some of the terminology is openly racist or sexist. For example, "male and female connectors" in electrical engineering; "master and slave servers" in IT; "FAQs" in computer science; and many others. Even in current workplace, there are practices that undermine underrepresented groups, such as lap dancers at Yahoo's hacking conference in 2009. Fortunately, after many complaints, Yahoo publicly apologized; however, the Internet was full of blogs in protest of the apology [13].

It is very important to use the proper language in the classroom. Students are watching their instructors as role models. If we see inappropriate terminology (for example, on a web page shown in class), we mention it and say why it is in inappropriate and what the alternative acceptable terms might be. For example, "master and slave server" should be shunned in favor of "primary and secondary server." We make sure to use appropriate terminology on all of our teaching materials. We also observe that students appreciate pointing out the reasons for changing the terminology. Students do appreciate when their identity is respected.

It is very important not to use any inappropriate expressions or language in the classroom. For example, it can happen that a capable female student answers questions that males did not know how to answer, and their response can be to laugh at the female. This kind of behavior has to be stopped right away, by clearly stating that laughing at others is unacceptable. Also, any offered suggestions that are inappropriate have to be rejected, gently yet firmly. For example, if students are asked to provide an example of clothing, and someone offers "crotch protector," we gently ask for a different and better suggestion, such as a T-shirt.

Visually, it is important to wear appropriate clothing and accessories that convey professionalism and also do not undermine the humanity of the person, and to provide a good role model for everyone. Technical fields have the reputation for being "geeky." There is a negative stereotype associated with geeks, as isolated white males [23]. It is important that the instructor looks like a "normal" person who has a life. For example, female students examine their female instructors, thinking: “Do I want to look like this when I get older? What does it mean to be a female technical professional? Do I have to become “one of the guys” and lose my femininity?” However, there are many different role models for “femininity” and any other flavor of minority “look” that students can choose from. Thus, it is important that the instructors present themselves in a genuine yet professional way, allowing students the space to be themselves as well, within the constraints of professional expectations. For example, Hawai’ian students of both genders may like to wear flowers in their hair, African-American students may like to wear braids, and female students may like to wear visible nail polish.

The stereotypical image of a professor is someone who is serious, business-like, and rather bland; broadly speaking, the society tends to think of professors through the typical stereotype of a “white bearded old guy” i.e. we think of the typical technical stereotype of middle-aged white male. Typically, this image also often involves suit and tie, or khaki pants and a dress shirt with dress shoes. Unfortunately, this image does not accommodate the alternative possibilities, both on the inner and outer planes. Furthermore, “business-like” does not allow for spontaneity, artistic expression, or too much humor, fun and happiness. If we are going to attract students and retain them, it is suggested to change these stereotypical views [22] [24]. We find improv comedy exercises, such as passing an imaginary beach ball to each other in class, an excellent ice breaker, mental refreshment and bonding tool, especially after long periods of intense focus on technical concepts.

7 Valuing Diverse Perspectives

Modern STEM curriculum is typically taught using examples from the Western world. However, there are many other perspectives. For example, the field of ethnomath uses mathematics from various cultures other than Western. For example, African cultures are known for using fractals in daily life, Polynesians navigated with the stars, and Native Americans use elaborate geometry. It is important to introduce such examples, because it gives students validation of their origins and a sense of value, competence and scientific ability. Many examples and references of this approach are shown in [21].

8 Results

In our experience, students can successfully collaborate and co-create in highly diverse environments. In this paper we present various strategies for increasing the sense of bonding in the community, giving students the voice, and achieving teamwork environment in classroom. Our classrooms are highly interactive, where students of all races, genders, ages, and abilities work together. Informal interviews with our colleagues teaching at institutions on the continental U.S. held at the NSF Engineering Ideas Lab in March 2014 indicate that this situation might not be as common as we take it for granted locally. We hope that our experiences can help to advance the cause of diversity. Further research needs to address accomplishing the fine line of co-creation rather than top-down approach, even in “entitled” environments.

9 Conclusions

This paper presents our experiences in valuing and encouraging diversity in our own classrooms. Each faculty can contribute very much towards these goals through their own classroom and other interactions. Potentially, we can help students achieve their academic and life goals, help our institution to attract, retain and graduate competent graduates, and help provide workforce able to thrive in the very diverse technical environment of today.

10 References

[1] National Academies of Science, "Rising Above the Gathering Storm, Revisited: Rapidly Approaching Category 5," National Academies Press, 2010.

[2] PCAST, "Engage to Excel: Producing One Million Additional College Graduates with Degrees in STEM," February 2012. [Online]. Available: President’s Council of Advisors.

[3] National Research Council of National Academies, "Discipline Based Education Research: Understanding and Improving Learning in," 2012. [Online]. Available: http://www.nap.edu/openbook.php?record_id=13362.

[4] National Science Foundation (NSF), "NSF13-126 Common Guidelines for Education Research and Development," September 2013. [Online]. Available: http://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf13126.

[5] D. M. Jawaharlal, "The Blog: Why is STEM Boring," 11 April 2011. [Online]. Available: http://www.huffingtonpost.com/dr-mariappan-jawaharlal/why-is-stem-boring_b_853107.html.

[6] J. M. Tkaczyk, "HIRE education: STEM and the transportation industry," 21 8 2013. [Online]. Available: http://www.seenmagazine.us/articles/article-detail/articleid/3259/hire-education-stem-and-the-transportation-industry.aspx.

[7] H. Huhman, "STEM Fields And The Gender Gap: Where Are The Women?," 20 6 2012. [Online]. Available: http://www.forbes.com/sites/work-in-progress/2012/06/20/stem-fields-and-the-gender-gap-where-are-the-women/.

[8] Federal Reserve Bank of Dallas Research Publications, "On the Record: PISA Results Shed New Light on U.S. Education Debate," First Quarter 2011.

[Online]. Available: http://www.dallasfed.org/research/swe/2011/swe1101e.cfm.

[9] OECD, "Programme for International Student Assessment (PISA)," 2013. [Online]. Available: http://www.oecd.org/pisa/.

[10] OECD, "PISA 2012 key findings," 2012. [Online]. Available: http://www.oecd.org/pisa/keyfindings/pisa-2012-results.htm

[11] Deborah Netburn, "Generation i? 40% of teens say they have an iPhone," October 2012. [Online]. Available: http://articles.latimes.com/2012/oct/10/business/la-fi-tn-teen-iphone-20121010.

[12] CNN, "American Generations Through the Years," May 2011. [Online]. Available: http://www.cnn.com/interactive/2011/05/living/infographic.boomer/.

[13] J. Mick, "Yahoo "Sorry" For Offering Lap Dances at Developers Event," October 2009. [Online]. Available: http://tech-beta.slashdot.org/story/09/10/22/1232225/yahoo-offered-lap-dances-at-hack-event.

[14] R. N. Charette, "The STEM Crisis Is a Myth," August 2014. [Online]. Available: http://spectrum.ieee.org/at-work/education/the-stem-crisis-is-a-myth.

[15] L. Rosen, "The Truth Hurts: The STEM Crisis Is Not a Myth," September 2013. [Online]. Available: http://www.huffingtonpost.com/linda-rosen/the-truth-hurts-the-stem-_b_3900575.html.

[16] U.S. Army, "The Creation of a Corps of Artillerists and Engineers," February 2014. [Online]. Available: http://usaes.armylive.dodlive.mil/.

[17] G. T. Lord, "History of Norwitch University," 2014. [Online]. Available: http://www.norwich.edu/about/history.html.

[18] B. Aronson, "On the Records: Texas 1 of 5 "Minority-Majority" States," Texas Tribune, February 2012. [Online]. Available: http://www.texastribune.org/2012/05/17/on-the-records-majority-texas-minority-races/.

[19] R. Atkins, "America Desperately Needs More STEM Students. Here's How to Get Them," Forbes Leadership Forum, July 2012. [Online]. Available: http://www.forbes.com/sites/forbesleadershipforum/2012/07/09/america-desperately-needs-more-stem-students-heres-how-to-get-them/.

[20] M. Dostis, "Degree alone not enough to prepare grads for workforce," US Today College, October 2013. [Online]. Available: http://www.usatoday.com/story/news/nation/2013/10/31/more-than-a-college-degree/3324303/.

[21] M. Barjaktarovic, "Teaching Mathematics and Programming Foundations Early in Curriculum Using Real-Life Multicultural Examples.," in *The INternational Conference on Frontiers of Education: Computer Science and Computer Engineering (FECS'12) (pp.78-84).*, Las Vegas, 2012.

[22] C. Overall, "Professorial stereotypes: We need to create a more realistic image of ourselves," University Affairs, April 2010. [Online]. Available: http://www.universityaffairs.ca/professorial-stereotypes.aspx.

[23] L. Kendall, "Geeks May be Chic, but Negative Nerd Stereotype Still Exists, Professor Says," Graduate School of Library and Information Science at University of Illinois, 2019. [Online]. Available: http://www.lis.illinois.edu/articles/2009/03/geeks-may-be-chic-negative-nerd-stereotype-still-exists-professor-says.

[24] N. Romanni, "UCLA professor seeks to break research stereotypes," University of California Daily Bruin, October 2013. [Online]. Available: http://dailybruin.com/2013/10/04/ucla-professor-seeks-to-break-research-stereotypes/.

[25] NSF, "NSF 14-033 Dear Colleague Letter: Preparing Applications to Participate in Phase I Ideas Labs on Undergraduate STEM Education," NSF, January 2014. [Online]. Available: http://www.nsf.gov/pubs/2014/nsf14033/nsf14033.jsp?WT.mc_id=USNSF_25&WT.mc_ev=click.

[26] D. M. Jawarhalal, "The Blog: Why is STEM Boring," The Huffington Post Education, April 2011. [Online]. Available: http://www.huffingtonpost.com/dr-mariappan-jawaharlal/why-is-stem-boring_b_853107.html.

[27] J. Tkaczyk, "HIRE education: STEM and the transportation industry," South East Education Network (SEEN), August 2013. [Online]. Available: http://www.seenmagazine.us/articles/article-detail/articleid/3259/hire-education-stem-and-the-transportation-industry.aspx.

[28] H. R. Huhman, "STEM Fields And The Gender Gap: Where Are The Women?," Forbes , April 2012. [Online]. Available: http://www.forbes.com/sites/work-in-progress/2012/06/20/stem-fields-and-the-gender-gap-where-are-the-women/.

[29] NSF, "Women, Minorities, and Persons with Disabilities," National Center for Science and Engineering Statistics; NSF Directorate for Social, Behavioral and Economic Sciences, 2013. [Online]. Available: http://www.nsf.gov/statistics/wmpd/2013/pdf/nsf13304_digest.pdf.

A Framework of Preparing Minority Students in Computer Science for the Industrial Career

Yujian Fu
Department of Electrical Engineering & Computer Science, Alabama A&M University, Normal AL, U.S.A.

Abstract— *The lack of preparedness of Computer Science graduates for a professional career is a common complaint raised by industry practitioners. The career progression of many new graduates is severely impacted due to the lack of problem solving capabilities and well rounded skills. For example, some of the technically stronger graduates lack communication and managerial skills and vise versa. Industry based capstone projects, incorporated as a part of an undergraduate degree, are a well accepted means of preparing students for their professional careers. This paper using a curriculum design of object oriented implementation class of graduate students at the Alabama A&M University with engaging in industry based projects at the first year of their degree to illustrate how to increase student motivation, improve software development capability, increase problem-solving capability and speed up the skills and knowledge of engineering discipline. The project developed in the curriculum aims at providing students a real-world software engineering development experience, improving student programming skills and project management capability and object oriented design concepts, and preparing them for industry. At the end of semester, we observed a successful achievement of these projects in giving aspects of evaluation. We believe this achievement is due to the practice on the object oriented programming language and project design framework. Finally, we believe this approach can be extended to any other objective problem solving framework, focusing on managerial, engineering and science classes.*

Keywords: Problem solving skill, capstone project, LEGO mindstorm, rentention

1. Introduction

Helping under-represented populations pursue STEM (science, technology, engineering, and mathematics) careers is a challenge issue now. Research shows an expanding ethnicity gap for Americans pursuing STEM careers. A new report from the National Action Council for Minorities in Engineering (NACME) reveals that the number minority students pursuing STEM degrees and careers has flattened out or even declined in recent years with the increasing requirement of workforce in IT professions [24]. Typically in Alabama A& M University, we have observed 66% drop of graduation rate from 2005 to 2013 for undergraduate students, and 29% drop for graduate students in computer science department. This symptom is due to several reasons. Loss of attraction of the major is one of the main reason. What students have learned are not useful and what are expected in industry are not taught in class – there is a gap between classroom teaching and career marketing. To solve this problem, encourage and support minority students in pursuing STEM careers, we need to a) promote classroom teaching in tangible and real-life oriented ways and b) promote fun ways to explore STEM interests and attract more students [26].

Real world projects and capstone projects are well accepted by industry practitioners, professional bodies and educators as necessary for providing practical experience to science and engineering students to prepare them for a professional career [11]. Such projects aim to consolidate the theoretical knowledge gained through subjects in related areas by practical application. Real world projects are particularly important in Object oriented design and programming course, Software Engineering (SE), and more advanced courses (e.g., software testing) in graduate level in minority institution, since most of the theoretical concepts in these courses are abstract in nature, many students are not ready for professional career preparation and, therefore, are hard to understand without adequate hands-on experience. Therefore, it is common for computer science courses to include capstone projects âĂŞ in some cases real-life software development projects âĂŞ as a part of both the undergraduate and graduate degree.

The curriculum of Master's of computer science degree at the Alabama A&M University includes both software engineering and object oriented programming as core courses. In addition to fundamental computer science subjects these courses cover both theoretical aspects, as part of their degree, students engage in year-long, industry-based projects. These projects are aimed at preparing students for their professional careers, as competent software engineers, through a real-life project experience. It is a well accepted fact that a competent software engineer requires a wide variety of skills such as managerial, engineering, team working and communication [2], [23]. One of the important learning outcomes we are interested in achieving from these project experience is the right balance between the depth and the breadth of problem solving skills. Over the past years we have tried various schemes for bringing about these learning outcomes with varying degrees of success.

In addition, it is worth to observe that computer science (CS) has lost its appeal, and robots can help find it. Even as computing has invaded every aspect of our lives, CS as a field of study is often seen as disconnected from these same lives. This is a key issue especially in minority institutions [31]. There over 40% students dropped from CS major in the past 20 years [33], [12], while there are over 60% students dropped from cs in AAMU, comparing to the increasing of CS jobs in the past. So to reestablish the connection, the Institute for Personal Robots in Education (IPRE, www.roboteducation. org) is developing a personal robot, software, and curricula to help teach introductory CS courses. Imbued with the proper pedagogical philosophy and training, it can help make CS more personal. Though there are islands of hope, overall, fewer students are enrolling in CS courses, staying with them, or moving into industry because we have washed them out of the classroom and the pipeline.

We believe that an objective framework built on the industrial oriented project (in our curriculum is robot project) to guide student learning is an important element for bringing about the expected learning outcomes through projects. The absence of such a framework tends to leave the students unguided, resulting in students not getting the right balance between the breadth and the depth of skills. Based on the experience gained from teaching project based subjects over several years, we have developed a skill-based objective framework, focusing on managerial, engineering and personal skills. In this paper, we present this framework and share our experiences of using it.

The rest of the paper is structured as follows. Section 2 introduce background of framework in two aspects and an overview of computer science education using robot projects. Section 3 presents the problem solving skill based framework with problem statement. Section 4 introduces the project design and observings in the problem solving based framework. Section 5 describes our early experiences of using the new framework and results analysis. Finally, section 6 concludes the paper.

2. Background

In this section, we will discuss the related works of problem solving and the basis of cooperative learning. Finally, an introduction of CS education in AAMU was described.

2.1 Problem Solving

Problem solving skills have been typically discussed in the engineering and science education [18], [27], [25]. Because the multidimensional requirements of the engineering program requires students to be able to handle diverse problems using multiple skills and knowledge. Inspired from engineering and science strategies, Several works from computer science education are established in the introductory or senior curriculum and program [9], [19]. A recent work of open-ended problem solving skills [21] used in the fluid engineering was discussed. In this work [21], a three step process of problem identification & formalating, skills teaching and data collection and analysis is presented. This paper have a definition of problem solving skills for each sample question. An alternative approach to problem solving skills was discussed in the work of [3] by Allan and Kolesar. Their approach is to introduce a preliminary course in general problem solving before teaching programming concepts. However, lack of problem solving capabilities among minority graduates is a serious problem raised from industry while was paid less attention by educators and researchers. Due to lack of background, inefficient learning, and other issues, minority students does not have motivation and therefore loss the interesting [32] in computer science. This has been reflected in the retention issue among minority institutions [30], [22]. A data from AAMU has shown up to 47% drop from sophomore to junior in computer science major [14]. How to improve student's problem solving capability in computer science is a key issue for student's career development and industry satisfaction.

Commonly, robots are used in introduction to computer science courses. Many institutions have introduced robot concept and implementation to the classes. Lego Mindstorms have been adopted for the introductory computer science course in George Washington University to teach problem solving using computers and applications of computer science to the real world [20]. Likewise, Lego Mindstorms and the LeJOS system were used in the introductory computer science course at Cansius College [5]. To teach basic computer science concepts such as control flow and the use of arrays, the United States Air Force Academy has studied using a previous version of Lego Mindstorms to teach an introductory course using Ada [10]. At some colleges, the visual programming language used in the Lego-supplied software to teach the concepts of computer science without making students who may be unfamiliar with a programming language to learn the syntax of a high-level language like C++ or Java, two of the languages that are commonly used in introductory computer science courses [6].

Robots have also been used in single-topic computer science courses. One of the areas that robots are most often used for pedagogical purposes is the field of artificial intelligence. At Williams College, robots are used to introduce students who may not be computer science majors to the concepts of artificial intelligence and behavioral programming through the use of robotic vehicles and a high-level language [8]. The City University of New York, Staten Island, also uses robots to teach artificial intelligence, but the course is tailored for third and fourth year computer science majors and requires a proficiency in C++ [13]. At the University of Salzburg, robots are used to teach embedded software engineering, covering real-time operating systems and communications protocols, scheduling, real-time programming, and code generation [16] (this link does not work). Villanova University has used Mindstorms robots to teach computer organization and operating systems concepts [17].

A common feature in both the introductory courses and the single-topic courses is that none of them require students to design a software artifact from a specification using software engineering practices. In the introductory courses, both for computer science majors and non-computer science majors, the projects are rather simple, like programming a robot to follow a line by using a light sensor, or pushing cans out of a dark area of a board [6]. While it is common for groups to be formed in introductory classes that use robots, the projects do not require, nor do introductory classes teach, software engineering concepts like specifications elucidation, design practices, or verification. Additionally, these classes do not attempt collaboration between groups. In the advanced, single-topic courses, the focus of the course is on a particular concept, and while it requires the use of sound software engineering practices, the course treats knowledge of such practices as a prerequisite [16].

2.2 CS Education in AAMU

Computer science department is one of the oldest department at Alabama state with the initiate of computer science education. With the past decades development, the department has both Bachelor and Master's degree with over 200 enrollment each year. The number is fluctuated with years. It is estimated to be around 30% to 70% African American students enrolled into computer science department Master's program roughly. There is a dramatic drop of enrollment of Master's students for the past 20 years. At the same time, the quality and recruiting of Master's students to computer science department has been a challenging issue since past years – it is necessary to maintain students numbers and improve student high level skills based on the undergraduate training background.

This problem-solving framework is especially designed for minority institutions with low enrollment of computer science students. Although Master's students are more motivated than undergraduate students, many students that do not have enough computer science background still feel very hard and boring about the current content. Secondly, the feedback from the industry of Master's student expects more solid skill and problem solving capabilities for graduate students than undergraduate students. Finally, for the course we teach, it is expected to have more interesting projects than just the theories and background knowledge. In addition to it, traditional projects cannot satisfy current needs and students' expectation. More challenging and interesting projects are highly desirable.

Due to the above typical characteristics of minority education in computer science and especially software engineering education, we propose this new interleaving framework to target problem solving capability of minority students in the software engineering education. Similar as the goal of the work in [15], we both prepare students to meet for industry requirements, which is one of the typical goals of education. Observing that it is hard to realize most of blocks in the framework of [15] in our school due to the above characteristics, we have two solutions on the table. One possible solution is improving our student quality by recruiting students with strong background and facing a dramatically drop down of student enrollment. Another possible solution is to improve our teaching strategy and adopt a new framework that works for the minority group. We believe that, after three years of implementation, our framework fits for the graduates of minority students with the above features.

3. Problem-solving based Framework

The main goal for introducing the problem solving-based learning framework is to provide objective guidance to ensure that the team based projects provide the expected software engineering skill outcomes with a proper balance between the breadth and the depth of problem solving skills. The new framework addresses the problems in the previous scheme while retaining some of the positive aspects. Project based problem solving framework includes four phases – problem identification, solution searching, solution carrying out, and validation & feedback. The main goal of the four phases is to provide stepwise management for the team based projects and instructions to the expected software engineering skills.

3.1 Framework

The Object oriented programming and design class at Alabama A&M University provides "various object oriented methodologiesâĂİ as part of its program capability and career support contribution [1]. In a similar vein, [7] also proposes a generic standard focusing entirely on the abilities and skills that graduates will develop as part of their bachelorâĂŹs and masterâĂŹs level programs. In line with this literature, our problem solving-based framework also identifies a detailed list of software engineering and development skills that computer science graduates must possess. The framework is categorized as four phases – problem identification, solution searching, solution carrying out, and validation & feedback. These are the sequential steps students choose to specialize in during the projects as described in the next section. A schematic representation of the skill-based framework is shown in Figure 1.

There are two facilitate streams around the four-phase. In other works, each phase is supported by pedagogical goal and studied by the research issues. The *problem identification* aims at selecting project topics from the real world

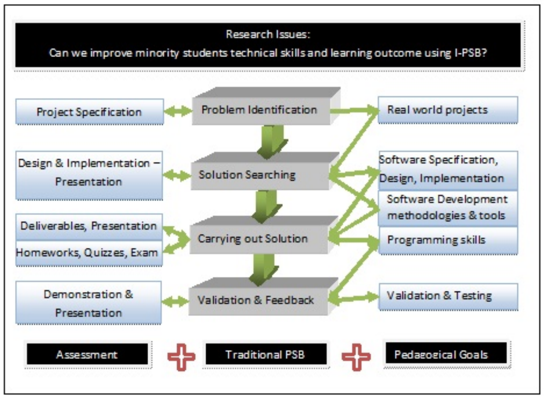

Research Issues:
Can we improve minority students technical skills and learning outcome using I-PSB?
Project Specification
Problem Identification
Real world projects
Design & Implementation – Presentation
Solution Searching
Software Specification, Design, Implementation
Software Development methodologies & tools
Deliverables, Presentation
Homeworks, Quizzes, Exam
Carrying out Solution
Programming skills
Demonstration & Presentation
Validation & Feedback
Validation & Testing
Assessment
Traditional PSB
Pedagogical Goals

Fig. 1: Framework

problem domain, design and adapt it to the class for the students. The research issues of this phase are:

- Can this project motivate students?
- Are students interested in the project or the series of projects?
- How does this project affect student's future career?
- Can this project attract more students?

The *solution searching* involves knowledge-based skills and application abilities. In this class, the software design, development methodologies and certain CAD tools are the fundamental purposes. The key issue in this phase is how to improve the student problem solving capability and help them to identify and choose the proper methods to realize the design. This is a critical phase for the capstone project, while iterated and incremental instruction will benefit student to make decision. *Carrying out solution* is a student-leading activity. Application of knowledge and methodology will be reflected with programming capability as well as the conceptual learning and critical thinking of the content. The project is designed as a team project so that student will have experience for collaboration work. Project management such as analysis, scheduling, and progress control are part of the goal in this phase. The evaluation of learning outcome does not include the results. Other assessment methods, such as required document, presentation and project demo, will help to assess the evaluation.

One of the key goals of the framework is to provide students with the right balance of problem solving capability from all domain areas. To facilitate this, students are asked to choose at least one problem domain from the provided projects and one development method from the provided methodologies. Students must fulfill all requirements of the selected project as these are aimed specifically at individual development. It is important to note that choosing certain development method does not constrain students from demonstrating their abilities in other areas.

3.2 Assessment

In addition to providing students with the industry oriented project and domain related topics, our problem solving-based framework also paves an objective path to assessment. The artifacts that submitted projects by students for their chosen topics in different pedagogical areas, along with related research issues and assessment criteria, form an objective and quantitative means of assessing students. The systematic assessment model also aids instructors to assess student performance on a fortnightly basis. Trends in student performance and the improvement shown over the course of the project can be extrapolated from systematic assessment and used for assigning an accurate final grade at the completion of the project. Such an objective framework lowers the risk of an instructorâĂŹs subjective views being induced into assessment, thereby providing a consistent grading scale across all project teams.

The grading is divided into four main components: project design component; project implementation component; project validation and presentation component; and demo component, each is weighted 25% of the final grade. The component grading criteria include: project specification, management, schedule, engineering, quality, and documents. The component weight are 10%, 15%, 15%, 25%, 15%, 20%, respectively. Although the project is assessed on the group based, each individual's contribution is also considered and expected to be identified based on the underlying skill. Such a distribution equips students with a well-rounded know-how of technical and non-technical software design, development and engineering issues.

4. Project Design and Observation

The projects are conducted and categorized as two groups – experimental group and control group. To reduce bias and increase the observation, all projects are controlled in the one domain in the year 2013 – embedded systems. The experimental group focus on the embedded system design and implementation with equipments, while control group works on the simulation of embedded systems using web technology or database. The projects of experimental group are designed on embedded system with the industrial based scenarios. All projects are modified based on the industrial advisory board members. The projects of study group includes the information system design and web design projects. There are 33% students chose the embedded system design and development based on their experience and background. Others chose information system design. A qualitative study of embedded system is given to participated and unparticipated students. The survey, including two aspects of the class project design and learning framework of the class, was given in 40 related questions at the end of the semester. We received very positive response from student survey. A roughly description of embedded system design project is given as follows.

4.1 Project Design

The industrial advisorary projects are based on the educational version of the Lego Mindstorms NXT system, which is available through Lego Education. It contains a programmable brick, three motors, one light sensor, one ultrasonic (sonar) sensor, one sound sensor, and two touch sensors. The programmable brick contains three output ports used to power the motors, and four input ports used to connect sensors. Legacy support is included via converter wires that connect to the NXT programmable brick and legacy 9V motors and Robotics Invention System sensors. [4].

The Mindstorms NXT brick uses a 32-bit ARM processor as its main processor, with 256 kilobytes of flash memory available for program storage and 64 kilobytes of RAM for data storage during program execution. To acquire data from the input sensors, another processor is included that has 4 kilobytes of flash memory and 512 bytes of RAM. Two motors can be synchronized as a drive unit. To give the robot the ability to "see,âĂİ the ultrasonic sensor, which is accurate to 3 centimeters and can measure up to 255 centimeters, and the light sensor, which can distinguish between light and dark, can be attached to the brick. A sound sensor that can be adjusted to the sensitivity of the human ear can be used to give the robot the ability to hear and react, if programmed, to noises. Finally, the two touch sensors give the ability for a robot to determine if it is pressed, or bumped, and react accordingly [4].

As a replacement for the standard Lego firmware, the LeJOS project has support for threading, arrays, recursion, synchronization, exceptions, non-generic data structures, standard data types, and input and output [29]. The LeJOS virtual machine supports much of the java.util package, but the data structures require that data be stored as type Object and then cast to a type that inherits Object. For input and output, both streams and sockets are available for use which is defined in the java.io package. For control purposes, the LeJOS platform supports the direct connection of Bluetooth-enabled GPS units for spatial location information and keyboards for the navigational control of a robot [28].

Table 1 lists three (3) project modules based on LEGO Mindstorms NXT.

4.2 Observing Problem Solving

Project management and the application of software modeling and testing concepts enabled students to observe the construction of the building a quality software system. To integrate *Problem Solving* into our curriculum, we have developed new projects, with a systematic assessment and evaluation procedure that explicitly specified in the syllabus. The only difficulty of this graduate class is that different students have different background in programming languages and skills. This is a typical situation in graduate courses in many HBCUs. Some even worst case is that a few students do not have enough programming skills and have to learn and compose programming concepts, to solve more abstract problems. In this special case, what the instructor has to do is to reduce the workload for the specific student and prepare make-up lectures. To make more students benefit from the all provided materials, all materials are available on line, as well as we insert a lecture dedicated to the demonstration of live problem solving.

5. Experience, Analysis & Discussion

All three projects can be finished within semester time, including new API learning, environment setting up, and programming and testing. There is one project was updated during break with more elaborated functions. The projects are very attractive to students. One student even got CO-OP from a company after introducing this project that he had been working on.

a) Experience: The computer science department of Alabama A&M University is an ABET credit and offers both Bachelor and Master degrees with enrollment of over 200 each year. Object oriented programming and design is offered in each fall semester following with software engineering methodology for first graduate students. Usually there are 50% to 75% African American students enrolled into graduate program. Students feel that it is hard to understand object oriented design such as UML (Unified Modeling Language). It seems it is not useful in many systems for the future especially in industry. Facing this problem and keeping the program outcome of computer science department in mind, the instructor has upgraded the class with real world project with consultant with industry board and discussing with education and curriculum department.

This paper is to show the experience of using the problem solving based framework, to improve student skills in the software development and design area and fit for the future

Table 1: Summary of modules based on LEGO Mindstorms.

Modules	Description
T7. Obstacle Removal	This module uses multiple sensors to achieve obstacle detection and removal. This project includes two tasks: programming Bluetooth-based remote control to the robot car and using ultrasound and touch sensors to detect and remove objects from the path. Students will practice object and class creation, method definition, event handling, control structures, streams, and array.
T8. Tracing and Yielding	This module traces a route with a light sensor, and yields obstacles with a touch sensor. Two tasks will be completed: route cues (e.g., strips) along the route are recognized, when stopped by an obstacle a new path is selected. Students will practice object and class creation, method definition, control structures, loop, and event handling.
T9. Automatic Navigation	The module implements a dynamic navigator that reaches the predefined destination. Path is decided based on images and signals from on-board camera and sensors. Shortest path is generated on-the-fly. Students will practice object and class creation, method definition, control structures, loop, and event handling.

career. The projects are developed on the LEGO Mindstorm NXT educational version. Three group projects are experimental group with industry oriented projects that is built on the LEGO Mindstorm NXT toolkit and implemented in Java/LEJOS. The projects were designed by consulting the industry advisory from companies. Some students had received notice from other company, and continuous communication between the company and students regarding to the internship and job openings are discussed. All students from three groups had shown much enthusiasm for their projects. With the continuation of the project, the students are more and more involved into the projects. In addition, some students (2) from control groups had expressed the idea of changing project to these robot projects.

b) Evaluation & Analysis: Our experience is that students are creative when they are allowed to be. If students feel that there is only one right answer to a problem, then they may feel restricted when experimenting on their own until they have been affirmed with selecting the right approach. When presented with the idea that multiple solutions can be correct, but have different tradeoffs, then they are more confident in exploring on their own. This could result in greater motivation to learn as well as understanding and thus manifest itself as improved learning outcomes for students.

Analysis of the course results using student experience surveys indicate that students feel more engaged with the course and the subject matter since the problem solving teaching has begun. Because the new framework has been in place for only one year, we are not in a position to perform a comprehensive evaluation. The qualitative and quantitative analysis are conducted at the end of each semester, feedback from students has been collected and the results are all positive. To reduce the bias, a set of background questions (question from 1 to 10) were given to students to evaluate their experience in the industrial, project design and implementation, programming on system platform, level and interests, etc.. Quantitative feedback (question from 11 to 30) was obtained through a set of student questionnaire at the end of each semester. Students are expected to assign a score ranging from 1 to 5 for each question, 1 being strongly disagree and 5 being strongly agree. Table 2 shows the averaged scores obtained for the subject in semesters 1 and 2, 2013 for evaluation questions applicable to the problem solving-based framework.

High scores for question 11 and 20 indicate that the main goal and research issues of the framework of achieving appropriate learning outcomes through providing clear expectations. Questions for 21 to 30 targets the class contents handling for the framework and organization. One of our main concerns when introducing the framework was of possible de-emphasis of the value of team work due to excessive focus on individual skills. A score of 5 out of 5 for question 19 clearly indicates that the team aspect still remained strong. The score for question 14 indicates that the intellectual interesting of the project stimulate and motivate student.

6. Conclusion

Real-world projects involving real-life project from industry clients are aimed at providing computer science graduates with the necessary skills required for their industry careers. Our experience in teaching computer science and software engineering subjects over the past years shows that the lack of an objective framework to guide student learning during such projects resulted in students not gaining the adequate breadth of software development skills that fulfill the program outcome of the department, and required by industry. This motivated us to develop a problem solving-based learning framework that aims to equip minority students with a well rounded knowledge of technical and nontechnical skills required to thrive in industry. In this paper we presented the problem solving-based framework and our experiences of using it for first year graduate students which are 70% African American students in software development projects at the Alabama A&M University in 2013 academic year. Overall, the application of the framework was a success from the perspective of students, staff and fulfilled its major purpose of providing students with an adequate breadth and depth of software engineering skills. In the coming years

Table 2: Evaluation of The Framework

Questions	Means
11. The project is effective and valid for student to learn the software design and development methodology.	5
12. The project is beneficial for student careers.	4.5
13. The work load of the project is appropriate .	4.0
14. The project topic is very interested and practical for the current class.	5
19. The team work in the project is successful and management of project provides us a lot of experience.	4.5
20. The learning resource and instruction of the project are available any time to the students.	4.5
30. The project topic is more interesting comparing to the peer projects in the peer classes.	5

we plan to improve the framework and its implementation to address some teething problems experienced and reported in this paper.

Acknowledgment

The authors would like to thank all reviewers for the kindly comments and suggestions on this work.

References

[1] AAMU. Aamu. Available from http://www.aamu.edu/gradstudies/, 2009.

[2] M. R. Ali. Imparting effective software engineering education. *SIGSOFT Softw. Eng. Notes*, 31:1–3, July 2006.

[3] V. H. Allan and M. V. Kolesar. Teaching computer science: a problem solving approach that works. *SIGCUE Outlook*, 25:2–10, January 1997.

[4] B. Bagnall. *Maximum Lego NXT: Building Robots with Java Brains*. Variant Press, 2007.

[5] D. Burhans, R. M. Meyer, P. Van Verth, D. Puehn, V. Steck, and J. P. Wiejaczka. Introductory computer science with robots. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1962–1963. AAAI Press, 2006.

[6] D. Cliburn. Experiences with the lego mindstorms throughout the undergraduate computer science curriculum. In *Frontiers in Education Conference, 36th Annual*, pages 1 –6, oct. 2006.

[7] A. J. Cowling, R. Court, and P. Street. What should graduating software engineers be able to do. In *Proc.16th Conference on Software Engineering Education and Training, IEEE Computer*, pages 88–98. Society Press, 2003.

[8] A. Danyluk. Using robotics to motivate learning in an ai course for non-majors. In *AAAI spring symposium*, March 2004.

[9] David J. Barnes, Sally Fincher, and Simon Thompson. Introductory Problem Solving in Computer Science. In *Proceedings of the 5th Annual Conference on the Teaching of Computing,*, pages 36–39, August 1997.

[10] B. S. Fagin, L. D. Merkle, and T. W. Eggers. Teaching computer science with robotics using ada/mindstorms 2.0. In *Proceedings of the 2001 annual ACM SIGAda international conference on Ada*, SIGAda '01, pages 73–78, New York, NY, USA, 2001. ACM.

[11] S. Fincher and M. Petre. Beyond anecdote towards real transfer: using other institutions' experience of project work. In *Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, ITiCSE '98, pages 86–89, New York, NY, USA, 1998. ACM.

[12] N. C. for Women and I. T. (NCWIT). Ncwit. Women and Information Technology by the Numbers. http://www.ncwit.org/pdf/BytheNumbers09.pdf, 2009.

[13] S. P. Imberman. Teaching neural networks using lego handy board robots in an artificial intelligence course. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, SIGCSE '03, pages 312–316, New York, NY, USA, 2003. ACM.

[14] A. IROSP. Aamu factors. Available from http://www.aamu.edu/gradstudies/.

[15] S. Karunasekera and K. Bedse. Preparing software engineering graduates for an industry career. In *Software Engineering Education Training, 2007. CSEET '07. 20th Conference on*, pages 97 –106, july 2007.

[16] K. Kirsch. Embedded software engineering course. University of Salzburg. http://cs.unisalzburg.at/ ck/teaching/EECS290O-Spring-2002.

[17] F. Klassner and C. Continanza. Mindstorms without robotics: an alternative to simulations in systems courses. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, SIGCSE '07, pages 175–179, New York, NY, USA, 2007. ACM.

[18] D. E. Knuth. Structured programming with go to statements. *ACM Comput. Surv.*, 6:261–301, December 1974.

[19] Z. Michalewicz and M. Michalewicz. *Puzzle-Based Learning: An Introduction to Critical Thinking, Mathematics, and Problem Solving*. Hybrid Publishers, May 2008.

[20] B. Narahari. CS 1: computer science orientation. The George Washington University School of Engineering and Applied Science http://www.seas.gwu.edu/ bhagiweb/cs1.

[21] J. R. Nikos J. Mourtos, Nicole DeJong Okamoto. Open-ended problem-solving skills in thermal fluids engineering. *Global Jounal of Engineering Education*, 8(2):189–200, 2004.

[22] K. E. Nugent, G. Childs, R. Jones, and P. Cook. A mentorship model for the retention of minority students. *Nursing Outlook*, 52(2):89 – 94, 2004.

[23] T. Nunan. Graduate qualities, employment and mass higher education. In *Proceedings of the HERDSA Annual International Conference*, 1999.

[24] C. on Professionals in Science and T. (CPST). Engineering degrees earned by under-represented minorities (urms). http://www.cpst.org/URM_Eng.pdf, 2006.

[25] G. Polya. *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press, 2004.

[26] S. Progress. Making robots personal. http://scienceprogress.org/2009/04/trower-interview-robots/, 2009.

[27] D. D. Riley. Teaching problem solving in an introductory computer science class. *SIGCSE Bull.*, 13:244–251, February 1981.

[28] L. Team. Nxj api. http://lejos.sourceforge.net/nxt/nxj/api/index.html.

[29] L. Team. Nxj technology. http://lejos.sourceforge.net/nxj.php.

[30] H. L. K. A. Tekian A, Han Y. Do underrepresented minority medical students differ from non-minority students in problem-solving ability? *Teach Learn Med.*, 13(2):86–91, 2001.

[31] J. Vegso. Interest in cs as a major drops among incoming freshmen. *Computing Research News*, 17(3), 2005.

[32] J. Wales. Achievement gap in the united states. http://en.wikipedia.org/wiki/Achievement_gap_in_the_United_States, 2011.

[33] C. S. E. Week. Computer science education week. http://www.csedweek.org/., Dec 2010.

Building a Linked-Courses Learning Community for Introductory Development Majors

Amber Settle and Terry Steinbach
School of Computing, DePaul University, Chicago, Illinois, USA

Abstract – *Student recruitment and retention in computing, particularly of underrepresented groups, remains a significant issue for post-secondary institutions. Linked-courses learning communities, in which students simultaneously register for two related courses and participate in co-curricular and extracurricular activities associated with those courses, have been shown to have positive effects on students' sense of belonging, student-to-student and student-to-faculty interaction, and student retention. We describe a project which develops a linked-courses learning community for students in several development-focused majors. Details about course selection, student recruitment, and program evaluation are provided. The project serves as a starting place for other computing educators interested in leveraging the benefits of linked-courses learning communities for student recruitment and retention.*

Keywords: CS1, digital divide, linked-courses learning communities, retention, underrepresented groups

1 Introduction

According to the Bureau of Labor Statistics, employment of computer programmers is expected to increase eight percent from 2012 to 2022 [5]. The Computing Research Association reports the number of new undergraduate computing majors has steadily risen the last five years [27]. Many institutions have implemented multi-faceted efforts in retaining those majors through graduation which indicates there is still a problem with retention and persistence of these new majors [2]. Dorn and Tew [9] cite several STEM researchers who have found evidence of a relationship between student perceptions and learning outcomes. In particular, students' achievement expectations and self-concept were good predictors of success; students' self-assessment of their ability impacts their decision to major in computer science; student beliefs affect how they learn new information and student experiences both inside and outside of the classroom can shape their beliefs. Schutle and Knoblesdorf [20] found that computing experiences affect a student's perception and attitude which enable or inhibit pathways into the field. Factors such as limited faculty-student encouragement and involvement with other students in the major have been shown to have a higher impact on underrepresented groups in computing, such as women [8].

An approach that has shown promise in improving students' sense of belonging is linked-course learning communities [12]. The origin of learning communities dates back to a six-year experiment at the University of Wisconsin in 1927 [12], and numerous models have been developed. In a linked-course learning community, students simultaneously enroll in courses from different disciplines or interdisciplines that are connected in content, purpose, and organization [6]. Learning communities intentionally restructure the curriculum to connect students and faculty [12]. The community is designed to provide students with an integrative and collaborative learning environment, with the aim of enhancing student achievement, reducing attrition rates, and increasing student and faculty enthusiasm [6]. Other issues that learning communities attempt to address are inadequate levels of interactions among students and between students and faculty [11]. Another term for such experiences is first-year interest groups, although FIGs typically involve more than just two connected courses [12], [22].

While linked-course learning communities have a long history, there is relatively little to be found in the computing education literature about them. There are articles that discuss linking content between technical courses, for example, by connecting user-interface and software-development courses [18]. There are also researchers who argue that designing and delivering sets of multidisciplinary sets of introductory computing courses is beneficial both for computing majors and for others who take those classes [14]. Further, there are learning communities in existence at several institutions that have a focus on computing or other STEM disciplines. Rochester Institute of Technology has what it calls Living Learning Communities in which students with common interests can live together and spend a year exploring their topic area of interest with faculty and through organized activities [19]. Syracuse University has a Science, Technology, and Math Learning Community in which students live together and receive academic support including advising and tutoring [23]. And Purdue University has a Computer and Information Technology Learning Community where students take common courses and participate in organized activities together [17]. But there are no published articles to be found discussing the logistics of creating a technology-focused linked-courses learning community or evaluating the effectiveness or impact of such a learning community. This is surprising given that recent recommendations for learning communities suggest that they be intentionally engineered around combinations of courses likely to be taken by students in clusters of majors [11]. Part of the problem may be that faculty who could create such communities lack administrative support or the ability to modify the courses to have integrative components as is recommended [22].

We focus here on the creation of a linked-courses learning community for computing majors. We describe a project currently underway at DePaul University to create a linked-courses community for two required first-term courses in three development-focused computing majors. We describe the courses chosen for the project and how they are being modified for the project. We also detail the process used to recruit students for the community, and outline the activities that will be a part of the co-curricular and extra-curricular components of the community. Finally we discuss the approach that will be taken to evaluate the impact of the linked-courses community on students' attitudes toward computing and on their study habits in programming courses.

2 Creating the Community

One of the first considerations when creating a linked-courses community is the target audience. As at many institutions, women and non-white men are underrepresented in the College of Computing and Digital Media (CDM) and particularly in the more technical majors found in the School of Computing. Our goal in creating this community is to improve the retention rates in these two populations. While the needs of the populations are different in some ways, they are placed together in the same community for practical reasons having to do with minimum class size. Efforts will be made to develop activities appropriate for both groups, and it is hoped that the mix of the two populations will produce some positive dynamics.

To ensure that students will be interested in joining the community two courses required in their major are the ones selected. It is also important to make sure that incoming freshmen are aware of the learning community and have sufficient opportunity to register for the linked courses. The creation of a linked-courses community is a challenging problem, in that students need to be aware of the community and what it can do for them when enrolling, which often occurs six months in advance of their first college courses. Reaching incoming freshman at that stage of the enrollment process involves coordination of efforts between staff at several offices in the university.

In the remainder of this section we describe the two courses chosen for the community, the approaches taken for student notification and recruitment, and the planned structure and activities associated with the linked courses.

2.1 Course choice and modification

The courses chosen for the project are required for each of the targeted majors to ensure that community is appealing for students and does not require undo effort to join. The majors targeted for this project are computer science (CS), computer game development: gameplay programming concentration (gaming: gameplay), computer game development: systems programming concentration (gaming: systems), and information assurance and security engineering (IASE). Each of these majors are ones in which students are expected to become strong programmers and have a significant number of courses focusing on application development in the curriculum.

Students declaring any of these majors have two choices for their first programming course. Students with little or no programming experience, which is a majority of students as expected from a recent nationwide study [10], are directed to take a Python programming sequence that makes no assumptions about prior experience. CSC 241: Introduction to Computer Science I and CSC 242: Introduction to Computer Science II focus on problem solving and application development in Python. The relatively simple syntax of Python allows the introduction of significant and interesting problems early in the course and provides the future developers targeted by the class with frequent opportunities for problem solving. Using Python to allow a focus on problem solving is something that other institutions have embraced [16]. The problem-solving approach in the class is enhanced by an extra 90-minute weekly lab section. The section is supervised by a graduate assistant, and during the session the students collaboratively solve programming exercises. The exercises are graded by the graduate assistant, and typically contribute 5-15% toward the course grade. Using a closed lab for programming classes has been shown to have positive benefits for retention, learning, and student satisfaction [3], [16].

Students who have earned a strong grade (above a B-) in at least one semester of a high-level programming language are directed to take a more accelerated Python course. CSC 243: Python for Programmers covers much of the material from CSC 241/242 but does so in a single quarter. The accelerated Python course focuses on a few applications that are particularly interesting when using Python, such as text processing, recursion, and HTML parsing, to ensure that students' procedural programming skills are strong. The class does not have a lab component associated with it. In-class activities include lecture and hands-on exercises, although the class is not held in a lab and not every student is required to bring a laptop.

Additionally, every freshman at DePaul is required to take a Chicago Quarter class. These classes are designed to acquaint first-year students with DePaul and the metropolitan community, neighborhoods, cultures, institutions, organizations, and people of Chicago. Each course has a Chicago-centric theme around which the academic topics are organized, and the courses come in two types: LSP 110: Discover Chicago or LSP 111: Explore Chicago. Discover Chicago courses meet for eight hours a day during the five-day week just prior to the start of classes, typically during the last week of August. The courses then have an abbreviated

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

schedule during the remainder of the quarter. Explore Chicago classes begin meeting when the regular quarter starts in early September, but meet for an extended class session. Each class is required to have multiple excursions into the city to see neighborhoods, institutions, and organizations relevant to the academic focus of the course. The course has standardized reading and writing requirements which are met using discipline-specific material designed by the instructor. The courses also have a "Common Hour," which addresses issues of transition for first-year students, including academic success skills and educational and career planning [7].

Students participating in the linked-courses community in development for this project will be enrolled during the autumn quarter 2014 in a reserved section of LSP 111 entitled "The Digital Divide." This Explore Chicago class explores the social issues surrounding access to information and communications technology (ICT). During the 1990s the digital divide was defined as access to the Internet and centered on racial and socio-economic differences. Today the divide has changed as technology has developed. Access has increased through the use of smartphones, but there are large differences in quality of connection, cost, and intent (entertainment versus empowerment). The course also considers the difficulties that people with disabilities face when trying to use ICT. A large focus of the course is the approach taken by the City of Chicago to address the metropolitan digital divide. The second co-author will be the lead instructor for the course, and she will be assisted by a student mentor from CDM.

Also during autumn 2014, the students participating in the linked-courses community will also be enrolled in a reserved section of CSC 241: Introduction to Computer Science I. The course normally has a prerequisite of college algebra, but this prerequisite will be waived for students in the community. The students will be given access to extra tutoring sessions and open hours to help them master the more mathematically challenging material in the course. If there are a sufficient number of students with programming experience in the linked-courses community, a second cohort of students in CSC 243: Python for Programmers will be created. The Python for Programmers class also requires college algebra which will be waived for community participants who will receive extra tutoring. Both the reserved sections of Introduction to Computer Science and Python for Programmers course will be taught by the first co-author. The lab section for Introduction to Computer Science will be taught by a graduate assistant who will be hired in June 2014. If possible that graduate assistant will also be hired for the extra tutoring sessions for the courses.

2.2 Student recruitment

The success of the linked-courses learning community will depend on recruiting sufficient numbers of students in the targeted population. With women and minority men representing a small portion of the overall enrollment in the CS, gaming: gameplay, gaming: systems, and IASE degrees, attracting as many as possible of the prospective students is crucial. In this respect the linked-courses community benefits from being implemented by faculty in CDM, a unit with a long-standing program for contacting prospective students as soon as they are admitted to the university [21]. Since 2004 prospective students have been paired with faculty members in the students' identified area of interest. Faculty members are directed to contact the students at least once and then to be available for questions from the students and their family. The program is unusual since undergraduate enrollment is centralized at DePaul and most units do not have contact with students until they agree to attend the university.

During the academic year 2013-2014, the authors have worked in conjunction with the CDM staff overseeing the prospective mentoring program. All students in the targeted population for the relevant majors and concentrations have been assigned to the authors as mentees. The students were initially contacted via email in early March 2014 with information about the linked-courses community. The authors also attended the in-person event for prospective students during March 2014. During the CDM Admitted Students' Day, prospective students and their families attend an afternoon-long information session and tour. The authors answered questions from their mentees about the community, CDM and DePaul as a whole. In April 2014 the authors reached out to students again when it became possible for admitted students to take their placement exams for DePaul University, taking the opportunity to remind them about the linked-course community.

Once the national May 1st acceptance deadline passed, students in the target audiences who identified one of the relevant majors or concentrations and submittd a deposit to DePaul were again contacted about joining the linked-courses community. Students who indicate interest will be given information about registration for the reserved sections of Explore Chicago and Introduction to Computer Science or Python for Programmers. Advising staff will assist them in registering for the appropriate courses before September 2014.

2.3 Planned activities

Although articles about computing-focused linked-courses communities cannot be found in the computing education literature, there are several learning communities in technology-focused programs in the U.S. including Purdue University [17], Syracuse University [23], and Rochester Institute of Technology [19]. These communities involve various co-curricular, extra-curricular, and communal-living opportunities. In developing the activities for the CDM linked-course community, the opportunities available to students in these programs were considered.

The CDM linked-courses community will involve at least one weekly co-curricular or extra-curricular activity. The activities will be organized and/or arranged by the authors, and as appropriate will involve other DePaul faculty and staff, staff from local technology companies and organizations, and DePaul student organizations. The final schedule for the activities is pending, but the following describes the current plan for the quarter.

The co-curricular activities will include:

- Special lectures, for example, a guest speaker from a local game development company
- Community-service projects
- Visits to technology companies and/or organizations, primarily in conjunction with the Explore Chicago course

The extra-curricular activities will include:

- Meals with faculty mentors
- A video game night, possibly in conjunction with DeFrag, the CDM student gaming organization
- Field trips to technology organizations that do not fit into the academic topic for the Explore Chicago class, including a visit to the 1871 incubator, companies that have been started by DePaul alumni and students, and companies associated with the CDM advisory board
- Pairing of advanced undergraduate mentors with the students
- Organized activities in the newly formed Physical Computing Lab in CDM

3 Evaluation

An important component of the linked-courses learning community project is the evaluation of its effectiveness. Taking the first-quarter Python course alone may have an impact on student attitudes and study habits as well as on student retention, since it employs techniques that have shown to be effective in retaining and engaging students such as Python as the teaching language, closed labs, and small class sizes [3], [16]. It is important to measure what impact the linked-courses community has independently of the experiences of students taking the first-quarter Python course.

We have developed a survey instrument to measure the impact of the project. The survey of student attitudes and habits administered for this work was based on previous surveys of student attitudes, including the Computer Science Attitude Survey [26], the USG Student Computing Survey [25] used in the Georgia Computes! Project [4], and the Science Interest Survey [13]. Because the survey would be administered in person during sessions of the programming class, brevity was important. Questions were selectively chosen from previous surveys and modified to meet the needs of this study, producing a survey with 33 multiple-choice questions and a single question that asked students to rank resources used to obtain new computing skills. There were also several demographic questions to start the survey.

Each of the multiple-choice questions are listed below, and students are asked to answer the questions using a five-point Likert scale of 5 = strongly agree, 4 = agree, 3 = neutral, 2 = disagree, and 1 = strongly disagree.

1. I plan to major in a technology-related degree
2. I am sure that I can learn programming
3. I am sure I can do advanced work in computer science
4. I think I could handle more difficult programming problems
5. I can get good grades in computer science
6. I have a lot of self-confidence when it comes to programming
7. I’m not good at programming
8. For some reason even though I work hard at it, programming seems unusually hard for me
9. Computer science has been my worst subject
10. It would make me happy to be recognized as an excellent student in computer science
11. I’d be happy to get top grades in computer science
12. If I got good grades in computer science, I would try to hide it
13. I’ll need programming for my future work
14. Knowing programming will help me earn a living
15. I will use programming in many ways throughout my life
16. Taking computer science courses is a waste of time
17. Once I start trying to work on a program, I find it hard to stop
18. I am challenged by programming problems I can’t understand immediately
19. I am easily frustrated by difficult programming problems
20. I do as little work in computer science courses as possible
21. I like talking with my friends about programming
22. I like to program in my spare time
23. I belong in the computing field
24. I feel isolated in computer science courses
25. I am part of a community of programmers
26. Computer science offers good financial opportunities after graduation
27. Computer science allows me to be creative
28. Computing offers diverse and broad opportunities
29. I have a lot of support that will help me to succeed in computer science courses
30. Computer science provides opportunities to make a difference in the world
31. I have a lot of friends who are interested in computing
32. My family is happy that I am taking computer science courses
33. I have had good teachers in my computer science courses

There is one additional question that asks students: “Outside of your classroom studies, what are your resources for learning/obtaining new computing skills? Mark all that you use and rank them in order of use. The one you use most should be ranked as 1, the one you use next often as 2, etc." The resources listed are: Friends/peers, Internet/web sites, professional organizations, self study, family members, tutors, faculty, and other (please specify). Students are instructed to leave a resource blank if they did not use it, so that only the resources utilized would be given a ranking.

This survey is in the process of being administered to every student taking the Introduction to Computer Science course during the 2013-2014 academic year, following procedures approved by the DePaul IRB. The results from this data will provide a baseline for understanding the impact that the first-quarter Python programming class has on student attitudes and habits. That information can then be compared to data gathered from students in the linked-courses learning community to gain a measure of the impact of the community on student attitudes and habits.

4 Discussion

The linked-courses learning community at DePaul addresses several important issues when trying to improve representation of underrepresented groups. In *Unlocking the Clubhouse*, Margolis and Fisher make the following recommendations for broadening accessibility to computing curricula [15]:

- Pay close attention to the quality of the student experience.
- Accommodate a wide range of computing experience among incoming students.
- Create a curriculum that reflects the many facets and impacts of computing.
- Establish structures for under-represented students to come together for support.

Prior to the creation of the linked-courses learning community, several of these recommendations were addressed in various ways. There are multiple paths for the introductory programming courses required by the CS, gaming: systems, gaming: gameplay, and IASE degrees, one for students with prior programming experience and one for students without any programming experience. Several student organizations exist to support underrepresented groups, such as HerCDM which is a mentoring and social group for female CDM students. There are a variety of degree programs at CDM, each with a slightly different focus on technology. In addition to the development-focused majors mentioned previously, there are degrees in information technology, interactive and social media, and more artistic areas like animation and digital cinema.

The linked-courses learning community builds on and extends these efforts, incorporating recommendations from the National Center for Women and Information Technology (NCWIT). NCWIT recommendations for improving female representation in computing [24] include:

- Recruiting strategically
- Retaining with student support
- Evaluating your efforts

The biggest contribution the community will make will be in enhancing the support structures for the development-focused majors. The students will share two classes, participate in co-curricular activities, and attend extra-curricular events as a group, which builds familiarity and a sense of belonging. It has been shown that student-to-student interaction is the most powerful predictor of students' intention to persist in the major beyond the first course [1], and the linked courses and required activities will promote this type of interaction. This is consistent with recommendations that student-to-student interaction be made a mainstream activity rather than an optional or purely extracurricular option [1].

Students will also be given extra tutoring and support to assist them in completing their first programming course, which should help build confidence and improve retention. Although the community is designed to last for a single quarter, students will have the option to enroll in sections together for follow-on classes, allowing them to create a cohort that lasts for several years. Formal evaluation of the program will be conducted to measure the impact of the learning community beyond the Introduction to Computer Science Python programming course.

5 Conclusion and Future Work

We present here information about the creation and evaluation of a technology-focused linked-courses community under development at DePaul University. The community links a general education course on the digital divide and a first-quarter programming class in Python, both of which are required in four development-focused majors in CDM. With the assistance of university staff, female and minority male students are being recruited from prospective students to join the community. Several co-curricular and extra-curricular activities are planned for the community, and students in the community will be provided with extra tutoring and support to improve their engagement and retention.

Evaluation of the implemented linked-courses learning community is important. The project will be evaluated using a survey instrument that is in the process of being given to every student taking the first-quarter Python course during the 2013-2014 academic year. Results from this baseline will be compared to the results from students in the community to understand the impact that the linked-courses community has on student attitudes and study habits. It is hoped that the results will show that the linked-courses community has a positive impact beyond what could be expected from taking the programming courses alone, but that is simply a conjecture at this point.

Although linked-courses communities exist at other technology-focused institutions [17], [19], [23], there has been little attention paid to the development or evaluation of such communities in the computing education literature. This article makes headway into this area by providing a starting place for other computing educators interested in designing and implementing similar interventions.

Future efforts on this project include the analysis of the baseline information about student attitudes and habits in the Introduction to Computer Science course, implementation of the learning community, and a comparison of the data from the participants in the learning community and the other introductory Python programming students not in the community. Results from the evaluation of the project will inform modifications of the linked-courses learning community in future academic years.

6 References

[1] Barker, L.J., McDowell, C., and Kalahar, K. 2009. Exploring Factors that Influence Computer Science Introductory Course Students to Persist in the Major. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, Tennessee, USA, March 2009).

[2] Beaubouef, T. and Mason, J. 2005. Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. *Inroads – the SIGCSE Bulletin* 37:2 (June 2005), 103 – 106.

[3] Boyer, K.E., Dwight, R. S., Miller, C.S., Raubenheimer, C.D., Stallmann, M.F., and Vouk, M.A. 2007. A Case for Smaller Class Size with Integrated Lab for Introductory Computer Science. In *Proceedings of the 38th ACM Technical Symposium on Computer Science Education* (Covington, Kentucky, USA, March 2007).

[4] Bruckman, A., Biggers, M., Ericson, B., McKlin, T., Dimond, J., DiSalvo, B., Hewner, M., Ni, L., and Yardi, S. 2009. "Georgia computes!": Improving the Computing Education Pipeline. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, Tennessee, USA, March 2009).

[5] Bureau of Labor Statistics, Occupational Outlook Handbook, Computer and Information Technology: Computer Programmers, http://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm, accessed April 2014.

[6] Cargill, K. and Kalikoff, B. 2007. Linked Psychology and Writing Courses Across the Curriculum. *The Journal of General Education*, 56:2, pp. 83-92.

[7] Chicago Quarter, First-Year Program, DePaul University, http://liberalstudies.depaul.edu/FirstYearProgram/ChicagoQuarter/index.asp, accessed April 2014.

[8] Cohoon, J.M. and Aspray, W.F. 2006. *Women and information technology: Research on underrepresentation*. MIT Press: Cambridge, MA.

[9] Dorn, B. and Tew, A.E. 2013. Becoming Experts: Measuring Attitude Development in Introductory Computer Science. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. (Denver, Colorado, USA, March 2013).

[10] Ericson, B. and Guzdial, M. 2014. Measuring Demographics and Performance in Computer Science Education at a Nationwide Scale Using AP CS. In *Proceedings of the 45th ACM Technical Sympoisum on Computer Science Education*. (Atlanta, Georgia, USA, March 2014).

[11] Johnson, K.E. 2013. Learning Communities and the Completion Agenda. *Learning Communities Research and Practice*, 1:3.

[12] Kellogg, K. 1999. Learning Communities. ERIC Digest. Retrieved April 2014 from http://eric.ed.gov/?id=ED430512.

[13] Lamb, R.L, Annetta, L, Meldrum, J., and Vallett, D. 2012. Measuring Science Interest: Rasch Validation of the Science Interest Survey. *International Journal of Science and Mathematics Education.* 10:3 (June 2012), 643-668.

[14] LeBlanc, M.D., Armstrong, T., Gousie, M.B. 2010. Connecting Across Campus. In *Proceedings of the 41st ACM Technical Sympoisum on Computer Science Education*. (Milwaukee, Wisconsin, March 2010).

[15] Margolis, J. and Fisher, A. 2002. *Unlocking the Clubhouse: Women in Computing*. MIT Press.

[16] Newhall, T., Meeden, L., Danner, A., Soni, A., Ruiz, F., and Wicentowski, R. 2014. A Support Program for Introductory CS Courses that Improves Student Performance and Retains Students from Underrepresented Groups. In *Proceedings of the 45th ACM Technical Sympoisum on Computer Science Education*. (Atlanta, Georgia, USA, March 2014).

[17] Purdue University, Community and Information Technology Learning Community, http://www.purdue.edu/studentsuccess/orientation/learning_communities/profiles/technology/computer_information.html, accessed April 2014.

[18] Reimer, Y.J. and Cassens, M. 2014. Perspectives on Co-linking Design and Development Courses in CS. In *Proceedings of the 45th ACM Technical Sympoisum on Computer Science Education*. (Atlanta, Georgia, USA, March 2014).

[19] Rochester Institute of Technology, Living Learning Communities, http://www.rit.edu/academicaffairs/advising/learning-communities/living-learning-communities, accessed April 2014.

[20] Schulte, C. and Knobelsdorf, M. 2007. Attitudes towards Computer Science-Computing Experiences as a Starting Point and Barrier to Computer Science. In *Proceedings of the Third International Workshop on Computing Education Research* (Atlanta, Georgia, USA, September 2007).

[21] Settle, A., Pieczynski, S., Friedman, L., and Kizior, N. 2013. Evaluating a Prospective Student Mentoring Program. In *FECS 2013: The International Conference on Frontiers in Education: Computer Science and Computer Engineering* (Las Vegas, Nevada, July 2013).

[22] Stevenson, C.B., Duran, R.L., Barrett, K.A., and Colarulli, G.C. 2005. Fostering Faculty Collaboration in Learning Communities: A Developmental Approach. *Innovative Higher Education*, 30:1.

[23] Syracuse University, Science, Technology, and Math Learning Community, http://lc.syr.edu/future-students/which-one/communities/science-tech-math.html, accessed April 2014.

[24] Thompson, L.D., Eney, C., Davis, R., and Grady, T. 2014. Recruit and Retain Women in Undergraduate Computing: Success Stories using Research-Based Practices. In *Proceedings of the 45th ACM Technical Sympoisum on Computer Science Education*. (Atlanta, Georgia, USA, March 2014).

[25] USG Student Computing Survey, http://dl.dropboxusercontent.com/u/2635522/USG%20Student%20Survey.pdf, accessed April 2014.

[26] Wiebe, E.N., Williams, L, Yang, K. and Miller, C. 2003. Computer Science Attitude Survey, Dept. of Computer Science, North Carolina State University, TR-2003-1, http://www4.ncsu.edu/~wiebe/www/articles/prl-tr-2003-1.pdf.

[27] Zwben, S. 2013. Computing Degree and Enrollment Trends. http://cra.org/govaffairs/blog/wp-content/uploads/2013/03/CRA_Taulbee_CS_Degrees_and_Enrollment_2011-12.pdf, accessed April 2014.

An Evolving Educational Advisory Role for Computer Science

Andrew J. McAllister, Genevieve Audet-Perron, Amanda Gilks, Debbie McAnany, and Rick Wightman
Faculty of Computer Science, University of New Brunswick
P.O. Box 4400, Fredericton, New Brunswick, Canada, E3B 5A3

Abstract – *Recent movements like Code.org seek to have every school-aged child taught the rudiments of computer programming. The goals of such initiatives are to recognize Information Technology (IT) skills as a foundational science in the new digital age, and to remove barriers for entry of young people into IT professions. While these goals are important, such change brings with it new needs for collaboration between post-secondary Computer Science educators and other stakeholders, which include teachers, government departments, post-secondary departments of Education, and industry associations. This paper provides lessons learned from experience with a variety of such interactions by members of the University of New Brunswick's Faculty of Computer Science. Immediate goals for continuing and enhancing these interactions are described.*

Keywords: Computer Science Education, Recruitment, Coding, Misperceptions of Computer Science

1. Introduction

Computer Science (CS) has historically been taught most often by post-secondary institutions. Now Computer Science courses are being introduced more and more as electives at the high school level (for example [1, 7, 10, 12]) and even in earlier school grades (for example [2, 6, 8]). We are also starting to see efforts to promote coverage of fundamental CS topics such as coding for all students in schools [11, 13]. This latter push is being popularized by movements like Code.org in the United States [5] and Code Kids in Canada [4]. The rationale for universal exposure to CS at a young age includes the following:

(a) Virtually everyone in our society is experienced as a consumer of information technology, but only a select few understand what is involved in creating that technology;

(b) Computational thinking is starting to be seen as a foundational competency in today's digital age. During their career, many professionals in a wide variety of disciplines will bump into the need to modify a spreadsheet, understand a macro, or tweak a website. Widespread exposure to CS basics will dramatically lessen the sometimes painful learning curve involved with such tasks, and will tend to improve the quality of the results; and

(c) Over the last decade or so common misperceptions of CS have resulted in lower enrollments in CS degree programs [3, 14]. Widespread exposure to CS concepts and skills is seen as one way to combat this trend.

Rolling out CS education in school systems is not, however, without its challenges. School teachers are taught what they need to know to teach traditional school curriculum topics, but relatively few teachers have a background in CS. Information technology (IT) professionals are currently blessed with such promising and well-paid career opportunities that teaching positions are often seen as less attractive. Then there is the question of what should be taught. Should the focus be on skills such as coding? Or are there other topics that would also provide significant benefits?

As school systems in numerous jurisdictions worldwide wrestle with these questions, we see a new and growing role for post-secondary CS educators as advisors to this process. This paper describes our preliminary experience with this role at the University of New Brunswick (UNB), and provides predictions for how this role is likely to expand in nature and scope in the near future. Our goal is to help other educators become more effective in this advisory role.

2. Experience with Students

The recruiting and transfer advising teams within UNB's Faculty of Computer Science bring a wealth of insight into the questions posed above based on years of experience dealing with students faced with the opportunity to choose whether or not to enter a Computer Science, Information Systems, or Software Engineering degree program. Our recruiters:

- Meet face-to-face with over 5,000 students in approximately 75 high schools each year, covering schools in the Canadian provinces of New Brunswick, Prince Edward Island, and Nova Scotia, as well as the U.S. State of Maine;
- Interact with middle school students at dozens of yearly math and science fairs; and

- Conduct "CS Unplugged" sessions with dozens of local elementary school classes each year. These sessions involve no technology, instead showing young children how they can use algorithms such as sorting values or keeping information safe, using hands-on physical resources like tool boxes with locks and our "super fun sorting mat."

All of these interactions involve both (a) an educational component where we strive to dispel common myths and explain the true nature of Computer Science to the students, as well as (b) considerable discussion where we hear about the students' perceptions, concerns, fears, goals, and preferences. This experience has led us to a number of insights, and has given us the opportunity to develop, refine, and gauge the effectiveness of intervention strategies for dealing with various challenges.

Insight #1: Fear is a significant factor when students choose what to study after high school.

One of the most frequent comments we hear from high school and potential transfer students is, "I heard CS is too hard." This even occurs when the student in question is already in a technology-oriented university program of study such as Engineering and is considering transferring to CS. Friends will advise against the switch, saying they are likely to fail out of the much tougher program.

Other fears also come into play, such as:

- "I don't want people to see me as geeky."
- "I don't want to be in classes filled with geeky people; I want to spend my time at university with people like me."
- "I'm a people person. I don't want to spend my entire career sitting behind a computer and never dealing with people."
- "I have no idea what Computer Science involves (except that I would be studying computers) and that uncertainty scares me."
- "I've never taken any CS courses, but I'm afraid I won't like all the strictly technical subject matter covered in them."
- "I've heard there are no jobs in CS anymore; I'm afraid I'll get to the end of my program and not be able to find a job."

A considerable percentage of high schools have very little idea what they would like to do for their careers, so discussions about their options often focus on what they believe they *don't* want to do. It doesn't take much fear and uncertainty to rule out IT from the conversation.

Insight #2: Familiarity with a subject area lessens students' fears of the unknown.

Middle and high school students typically gain repeated exposure to subjects such as math, English, sciences, history, and so on. As a result they tend to develop confidence that they can handle the work involved with learning in these areas, because they have been there and done that. For the large number of graduating high school students who have no particular career path in mind, this confidence often seems to tip them in direction of programs such as Bachelor of Science or Bachelor of Arts. The student may not have much of an idea of where such a degree will lead them, but we frequently hear that at least they are not afraid their studies will be "too hard."

One of the benefits of providing CS instruction as part of core school curriculums is that CS would become one of those familiar and therefore somewhat less scary subject areas. We believe, however, that CS courses in schools can and do sometimes fail to alleviate other fears. For example, a course that focuses solely on coding can actually reinforce fears that a CS degree and resultant career would be all about sitting behind a computer and dealing strictly with technology.

Insight #3: Understanding the true nature of CS and IT career paths seems more important than competency in CS subject matter when selecting a degree program.

We almost never hear a high school student say they lack the confidence to enroll in a CS degree because they feel they lack sufficient skills in coding or algorithm design in order to successfully begin the program. Given that CS courses are not required in high school (and are not required for applying to CS degree programs), most students seem to realize this lack of knowledge is not a barrier to entry. What we do hear is, "I like using computers," or "I like playing computer games so I thought CS might be something I like."

When asked what they think CS is about, however, the answer is almost always, "I'm not sure," "I don't really have any idea," or "The study of computers, I guess." Our experiences in this regard are consistent with the findings of more formalized studies such as [9].

Young people learn what doctors, police officers, lawyers, and a wide variety of other professionals do through a number of life experiences, including books, television, movies, and real-life interactions with these professions, for example by having an annual check-up with a family physician. These experiences place these professions in the consciousness of young people and increase the chances they will consider them as possible career choices. The IT profession, however, is somewhat of an invisible trade by comparison. Most young people are unlikely to wander into an IT professional's workplace. Scrum sessions and JAD workshops don't seem to be high on the list for inclusion in Hollywood films. Given this absence of information, the general public ends up having very little idea what IT is all about.

We have had good luck getting young people to understand the problem-solving nature of CS by using anecdotal stories, which we relay verbally to individuals or groups as the situation dictates. One favorite is about a computer scientist who visited a neonatal intensive care unit at a hospital and noticed a variety of electronic monitoring devices in use with each critically ill newborn. Collectively these devices were capturing a considerable volume of data on the status of each infant, however only a very small portion of the data values were being sampled, usually by a nurse or physician walking over to the machines and looking at the readings.

The computer scientist thought, "We should be able to make better use of all this data," and set out to create a software solution to analyze the data more completely. One result was that they were able to start detecting the presence of a particular type of infection up to a full day earlier, which of course means the infection is less well established and is easier to combat, with less impact on the health of the infant.

We explain this is what CS is all about – using information to solve problems with real-world importance. The fact that computers are often a part of the solution is not the point; such devices are simply our best current tools for storing and manipulating data into useful information in various forms.

We go on to explain that the same is true in other disciplines. Biologists use microscopes to study organisms, but biology is not the study of microscopes. Similarly, chemistry is not the study of test tubes, and CS is not the study of computers.

At this point in the dialog we usually see plenty of head nodding and "aha" moments happening. In many cases we have found that this level of insight is effective in turning young people on to the idea of becoming a problem solver who can inject real value into a wide variety of enterprises. Grover, Pea, and Cooper report similar success using a set of videos to explain the true nature of CS [9].

Insight #4: The use of online and self-directed learning seems more prevalent for high school courses with more technology-oriented subject matter, with sometimes less-than-optimal results.

Within the many high schools we visit each year, we have never heard of courses in English literature or history being offered online or via self-directed computer-based modules. There are a few examples, however, of "Broad-Based Technology" and computer programming courses that are offered in this manner.

Given how short the attention span of young people can be, it is our opinion that such a self-directed, technology-driven mode of instruction is often likely to be less effective than traditional in-class, face-to-face instruction provided by teachers. This opinion is backed up by considerable anecdotal evidence – stories we have been told of students who are keen to take CS at university, but change their minds after having a negative experience with a self-directed CS course in high school.

The primary reason why technology courses are taught this way in our local schools seems to be primarily due to the aforementioned lack of teaching personnel who are qualified to teach technology-related subject matter. For instance, in the approximately seventy schools we visit regularly, only three are fortunate enough to have a teacher on staff who is qualified to teach Java programming. It is no surprise, then, that our local school districts rely on "pre-canned" online subject matter in order to offer instruction across a larger number of schools.

3. Stakeholder Relationships

Figure 1 shows a number of the most important stakeholders involved in typical school systems, along with the most salient existing relationships (shown as black arrows).

The primary focus of this entire discussion is the teaching relationship between school teachers and their students. The hope is that appropriate skills and knowledge can be imparted to students, consistent with the three-part rationale presented in Section 1: developing creators of technology, developing computational thinking as a foundational competency, and enhancing uptake of CS as a profession.

This direct relationship between teacher and student is the focus of movements such as Code.org and Code Kids. The basic idea is that all students should learn the fundamentals of coding. In other words, these movements strive to influence what subject matter is presented to students in schools.

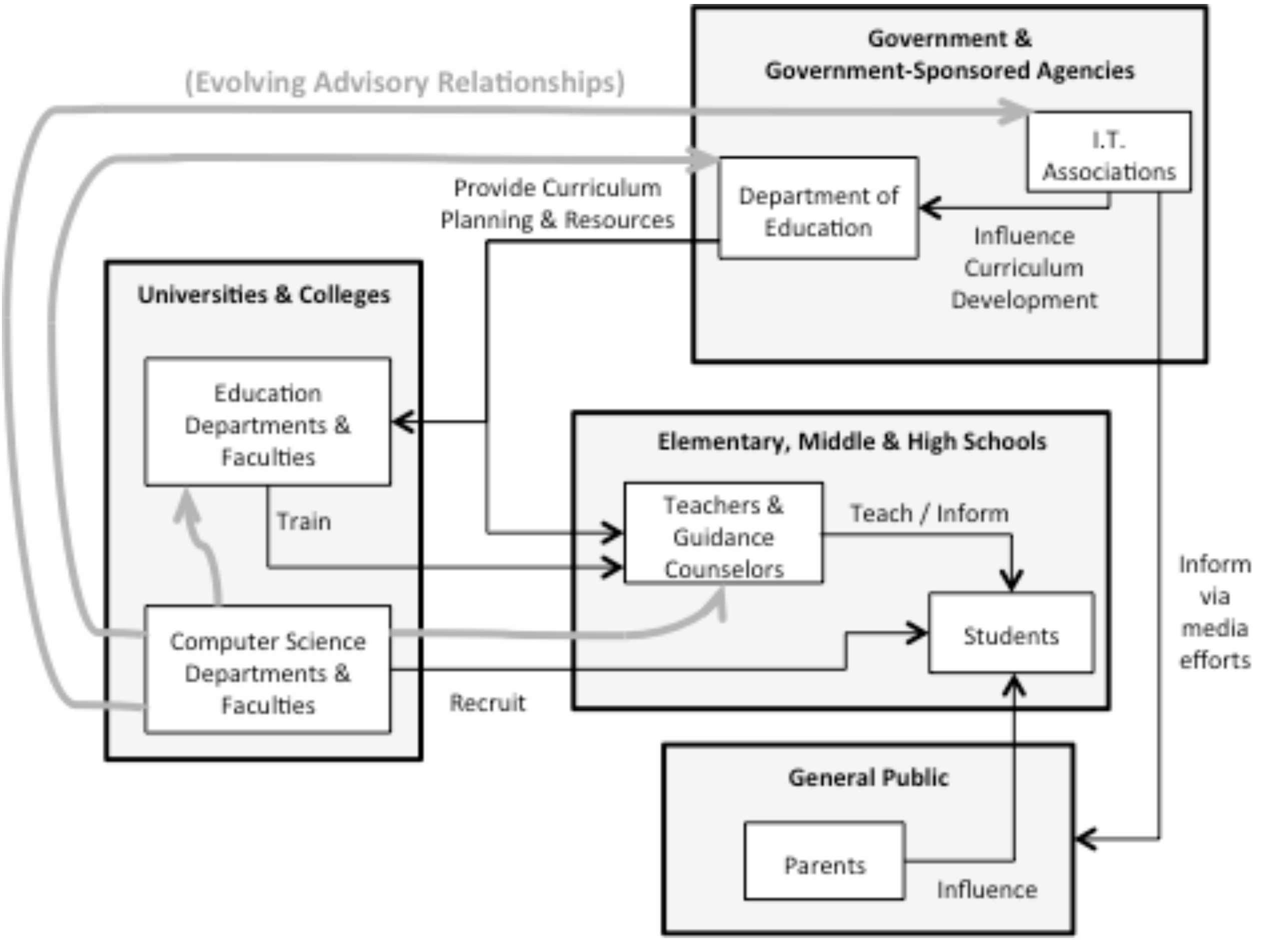

Government &
Government-Sponsored Agencies
(Evolving Advisory Relationships)
I.T. Associations
Department of Education
Provide Curriculum Planning & Resources
Influence Curriculum Development
Universities & Colleges
Education Departments & Faculties
Train
Elementary, Middle & High Schools
Teachers & Guidance Counselors
Teach / Inform
Students
Inform via media efforts
Computer Science Departments & Faculties
Recruit
General Public
Parents
Influence

Figure 1: Educational Stakeholders and Relationships

As Figure 1 shows, however, the impacts of such changes ripple through a number of stakeholder groups:

- New teachers must be hired or existing teachers must be re-trained to be able to teach new subject matter;
- Teachers are normally trained, for example, by Education departments or faculties at universities or colleges. To implement a new foundational science across all (or even many) school curriculums, the teacher training at this level would have to evolve; and
- It is common for school curriculums to be developed centrally by government Departments of Education and provided for teachers across a given jurisdiction to deliver; this is the case in our province. Changes to curriculums impact this activity.

A number of industry- and government-sponsored associations are also on board with the need to increase exposure to CS among school students. These organizations typically use a multi-pronged approach for information dissemination, including influencing school curriculums. For example, the Information and Communications Technology Council (ICTC) of Canada conducts the Focus on Information Technology (FIT) Program. Participating provincial jurisdictions align their high school curriculums with a set of IT-related goals defined by ICTC. High school students can voluntarily sign up to participate in the program, and receive a diploma recognizing their efforts in completing relevant courses.

In many jurisdictions, the governmental Departments of Education and university-level Education departments suffer from the same challenge as the schools themselves, namely a lack of fully trained IT resources who can effectively (a) develop curriculums, and (b) train the teachers. For this reason we believe it is becoming increasingly important for university CS faculty members to provide expertise by taking on new advisory roles, which are shown as thick gray arrows in Figure 1.

At the University of New Brunswick we have already undertaken this advisory role. For instance, each year we conduct a full-day workshop for the entire graduating class of UNB's Faculty of Education. We explain how teachers can implement computational thinking into courses as diverse as English and mathematics, and the day includes participatory exercises that prepare these teachers-to-be to do exactly that.

We have also begun working with the ICTC to influence the types of IT-related knowledge encompassed by the FIT Program. Our hope is to work together to ensure students are exposed to not only just IT skills, but also material that helps dispel the most common fears and misperceptions surrounding CS.

4. Looking Ahead

Despite our progress in working with various educational stakeholders, we recognize that an ongoing and evolving road lies ahead. We are committed to serving as an effective source of help and information for continuing development of CS-related subject matter in local and national school curriculums.

At the present time we are working on addressing the limited effectiveness of rolling out CS courses in middle and high schools due to a lack of teachers with the required IT expertise. The fundamental question is this; what can we collectively do to significantly improve the information making it to all school students without requiring large numbers of qualified IT professionals to act as teachers? How can we leverage the human resource base comprised of existing teachers to improve upon the current situation, based on only a reasonably modest investment of cost, time, and effort?

We believe the answer lies in looking beyond the current efforts to teach coding to all students. Such efforts are likely to take time to implement, especially given the well-recognized shortage of teachers with IT knowledge. In the short term, we would like to help large numbers of teachers in multiple disciplines become effective advocates for CS. We have already started doing this with newly graduated teachers, and we hope to do so with existing teachers as well.

The key to this initiative is to recognize three factors:

1. Teachers can educate students about the true nature of CS using anecdotal stories of the type described in Section 2 above. Videos can also be incorporated into such presentations;
2. The ability to do so does not require that teachers have any specialized IT skills; and
3. Teachers can be prepared to take on this role in as little as a single Professional Development (PD) day. (The challenge, of course, is to convince school system officials of the value of this role, and to obtain agreement to allocate the required PD days.)

We are determined to serve as agents of this change, and we encourage other university and college CS educators to explore opportunities to take on similar advisory roles within your own local educational systems. The potential positive impacts on our IT economy and our society are sure to be well worth the effort.

5. References

[1] M.A. Bernardo, J.D. Morris, Transfer effects of a high school computer programming course on mathematical modeling, procedural comprehension, and verbal problem solution, Journal of Research on Computing in Education, 26, 4, Summer 1994, pp. 523-536.

[2] P.S. Buffum, A.G. Martinez-Arocho, M.H. Frankosky, F.J. Rodriguez, E.N. Wiebe, K.E. Boyer, CS Principles Goes to Middle School: Learning How to Teach "Big Data", *Proc. SIGCSE'14: 45th ACM Technical Symposium on Computer Science Education*, March 2014, Atlanta, pp. 151-156.

[3] L. Carter, Why students with an apparent aptitude for computer science don't choose to major in computer science, *Proc. SIGCSE'06: Symposium on Computer Science Education*, Houston, March 2006, pp. 27–31.

[4] Code Kids, *http://www.codekids.ca*.

[5] Code.org, *http://www.code.org*.

[6] J. Denner, L. Werner, E. Ortiz, Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?, *Computers & Education*, 58, 1, Jan. 2012, pp. 240-249.

[7] W. Feurzeig, S. Papert, B. Lawler, Programming languages as a conceptual framework for teaching mathematics, *Interactive Learning Environments*, 19, 5, Dec. 2011, pp. 487-501.

[8] M.N. Giannakos, L. Jaccheri, R. Proto, Teaching Computer Science to Young Children through Creativity: Lessons Learned from the Case of Norway, *Proc. CSERC'13: 3rd Computer Science Education Research Conference*, Arnhem, Netherlands, April 2013, pp. 103-111.

[9] S. Grover, R. Pea, S. Cooper, Remedying Misperceptions of Computer Science among Middle School Students, Proc. SIGCSE'14: 45th ACM Technical

Symposium on Computer Science Education, March 5–8, 2014, Atlanta, pp. 343-348.

[10] J. Liebenberg, E. Mentz, B. Breed, Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT), *Computer Science Education*, 22, 3, Sept. 2012, pp. 219-236.

[11] H. Partovih, Transforming US Education with Computer Science, *Proc. SIGCSE'14: 45th ACM Technical Symposium on Computer Science Education*, March 2014, Atlanta.

[12] T. Paz, D. Levy, Introducing computer science to educationally disadvantaged high school students, *Research in Science & Technological Education*, 23, 2, Nov. 2005, pp. 229-244.

[13] E. Shein, Should Everybody Learn to Code?, *Communications of the ACM*, 57, 2, Feb. 2014, pp. 16-18.

[14] S. Yardi, A. Bruckman, What is computing?: bridging the gap between teenagers' perceptions and graduate students' experiences, *Proc. ICER'07: Third International Workshop on Computing Education Research*, Sept. 2007, Atlanta, pp. 39–50.

A 31-Year Record of CS Enrollment at a Single Urban Public Institution

Aaron Tenenbaum and Yedidyah Langsam
Department of Computer and Information Science, Brooklyn College of CUNY, Brooklyn, NY, USA

Abstract- *We present a 31-year record of undergraduate and Master's-level Computer Science enrollment at Brooklyn College, a comprehensive, urban institution in New York City. Data is presented for overall enrollment, separate undergraduate and Master's level enrollment, enrollment in CS0 and CS1, and in a course in computer applications. A number of "noise" factors (changes in the nature of courses, in enrollment policies, in course requirements and in technological developments) are also identified in the interpretation of the raw data.*

Keywords: computer science, enrollment, history,

1. Brooklyn College

Brooklyn College is a comprehensive unit of the City University of New York. The University consists of 19 constituent colleges within the City of New York, and is funded by the State of New York and tuition revenues. The University is the largest system of higher education in the country in terms of enrollment [1] and contains community colleges, 4-year colleges, comprehensive colleges, a law school and other specialty schools. The University has a Ph.D. Program in computer science, which is a consortial program among the various constituent colleges, headquartered at the University Graduate Center. The individual colleges of the University offer Associate, Bachelor, and/or Master's degrees in computer science and in associated computing disciplines.

Located in the borough of Brooklyn, Brooklyn College has had, over the 30-year period documented by this paper, between 14,500 and 17,500 students, of which approximately 2,000-3,000 were Master's students. The demographics of the student body are 32% White, 18% non-Hispanic Black, 12% Hispanic, 14% Asian, and 24% Other and Unidentified. The largest two undergraduate majors are Accounting and Business. Computing, in all its forms, represents the 7th largest number of majors in the College, after Business, Accounting, Education, Psychology, Health Science and TV Radio.

The College grants Bachelor and Masters degrees in Computer Science and in Information Systems, and a Bachelor's degree in Multimedia Computing and in Computational Mathematics. Table 1 indicates the date that each of these programs began, any departments whose courses constitute over a third of the program, and the current number of majors in each program. The Department of CIS consists of 26 full-time faculty, 15 adjunct faculty, and 3 graduate fellows. Full-time faculty members of CIS are also active in the University Ph.D. Program in Computer Science.

Table 1: Brooklyn College Programs in Computing

Program	Year Begun	Departments	# of Majors (Feb. 2014)
Bachelor's in Computer Science	1974	CIS	289
Bachelor's in Computational Mathematics	1972	CIS / Mathematics	9
Bachelor's in Information Systems	2006	CIS / Business	82
Bachelor's in Multimedia Computing	2006	CIS	43
Master's in Computer Science	1977	CIS	61
Master's in Information Systems	1989	CIS	66

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

2. Data Collection

Professor Frank Beckman was Chair of CIS in the period 1972-1985, and may be credited with establishing the department, putting it on a firm academic footing and making its offerings attractive to students. Professor Aaron Tenenbaum (one of the authors) was Chair during the period 1985-2009 and began collection of this data. When he became Chair, he retrieved enrollment figures for the previous three years, 1982-85. From that time on, he updated a table each semester with that semester's enrollment figures in order to report one facet of the state of the department to the faculty on a regular basis. Professor Yedidyah Langsam (the other author of this paper) continued this practice when he became Chair in 2009 until the present date.

Generally, the data was taken on the day that regular student registration for the semester ended, and may not account for drop-outs and late registrations after that date. However, since the data was taken at the same relative date each semester, the trends it displays are valid. No conclusions should be drawn from the absolute numbers, but only from comparisons (in terms of percentage increases and decreases) between different semesters and years and the trends they produce.

A student undergraduate FTE represents 15 credit-hours, while a graduate FTE represents 12 credit-hours. All charts are in terms of student FTEs, whether graduate, undergraduate, or a blend of both, as appropriate.

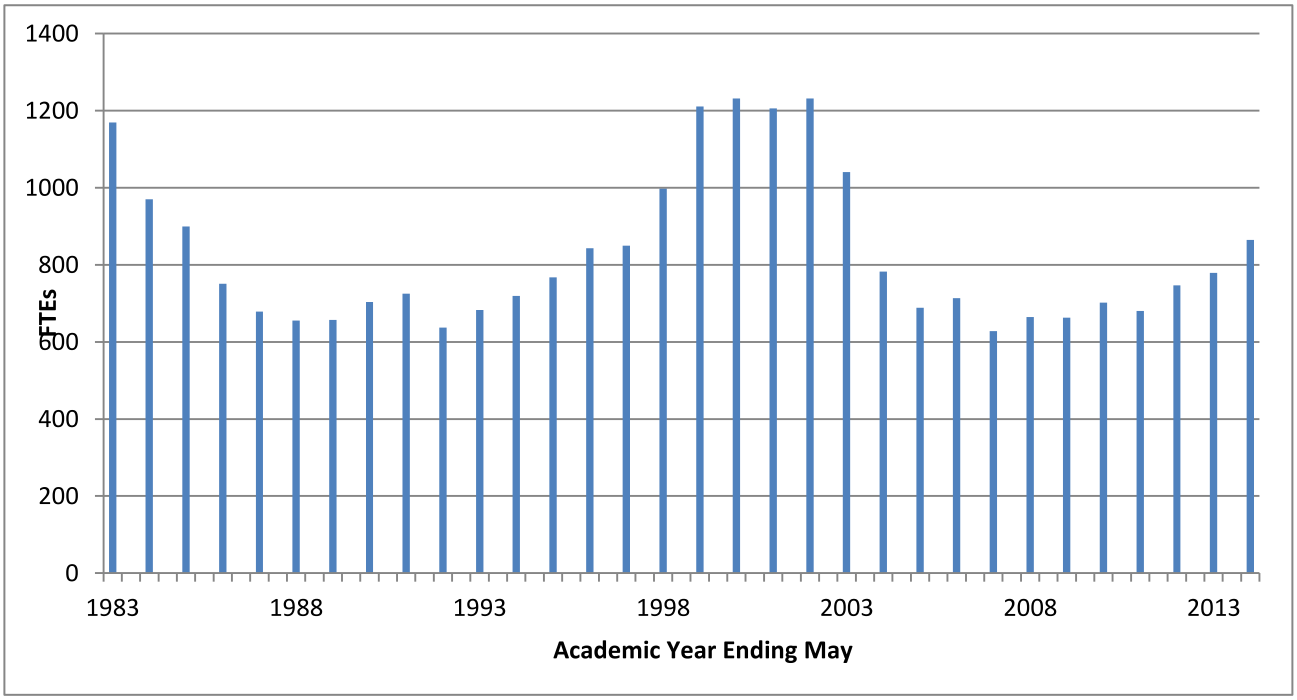
3. Total Undergraduate CS Enrollment

Chart 1 shows total annual Undergraduate Computer Science FTE enrollment at Brooklyn College from 1982-83 through 2013-14 (Fall plus following Spring). This does not include enrollment in CS0, which is part of the College Core Curriculum and is not required for the major. Information on CS0 will be presented later in the paper.

This chart is very similar to a chart that appears in a report from the Georgetown University Center on Education and the Workforce [2]. That chart was taken from NSF data on the percentage of Computer Science degrees among all bachelor's degrees granted and the percentage of CS majors in a UCLA survey of freshmen shifted forward by 4 years. This supports the notion that the Brooklyn data is a reflection of the national picture.

The lowest value on this chart (636.9 in 1992) is approximately 48% below the highest value (1232.0 in 2000), affirming wide swings in demand for computer science education. The chart also suggests that these swings are cyclical.

Chart 1: Undergraduate Computer Science FTEs



3.1 Causes for the Wide Swings

Many computer science departments were excoriated by their administrations in the mid-decade 2001-2010 for dropping enrollments, with the assumption that, somehow, the departments were to blame.

The fact that the trend was nationwide, which is well documented [3], indicates that the drop more likely was due to societal shifts: the national economy and job market, and the shift in the nature of computer skills that became necessary at that time (the end of the Y2K crisis, the advent of more user-oriented computing tools, and the change in programming and computing methodologies and modalities). We recall that high-school counselors were actively discouraging their students from seeking to major in computer science at that time. Another factor that may have caused this was the well-publicized forced retirements of experienced information systems workers [4] that occurred at that time as the workstation and server replaced the mainframe as the chief engine of commercial computing.

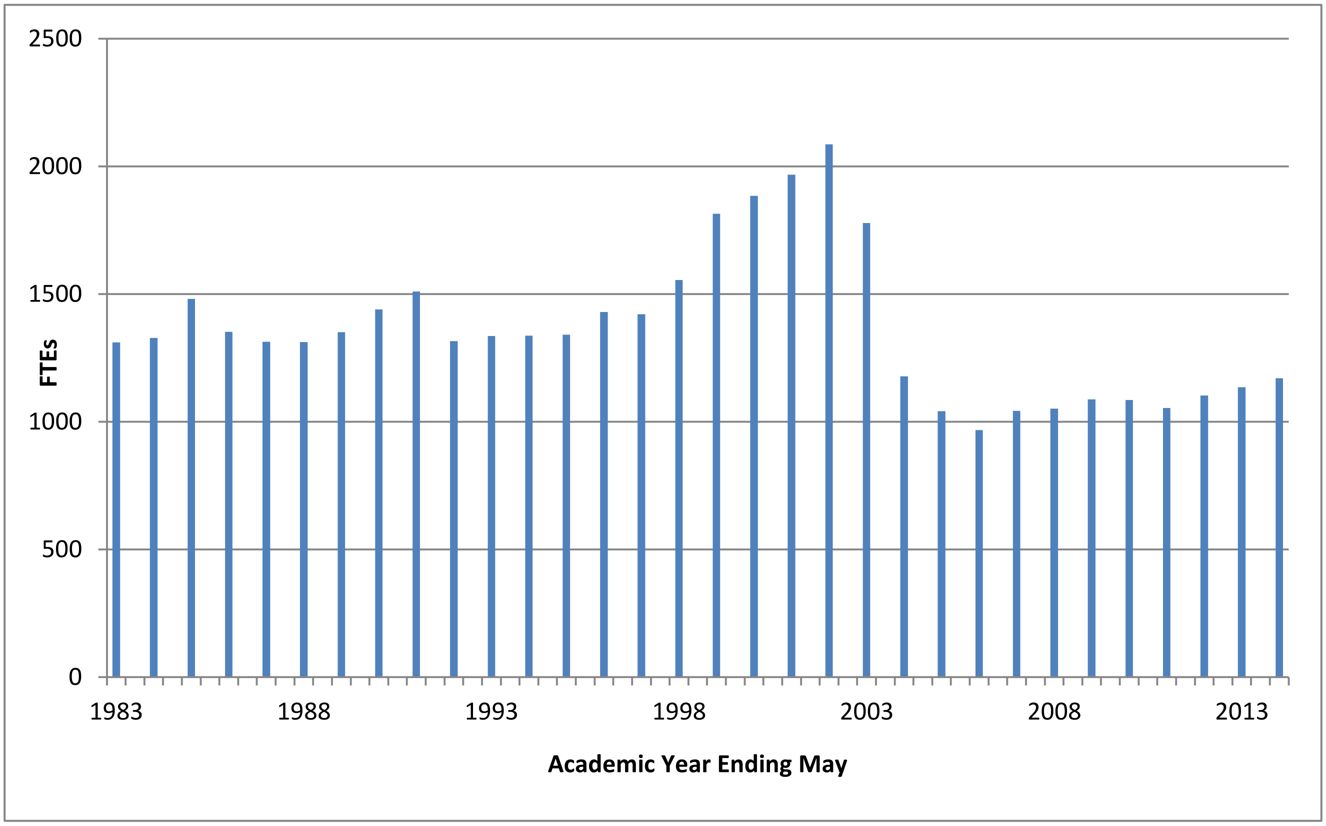
Now, however, there is a strong demand for computing skills again, much greater than the supply [5]. So the graph is again turning upwards as this fact becomes more generally known. A study of the precise factors involved in both the drops and the recoveries would be quite useful so that the future can be shaped (or at least predicted) based on the mistakes and events of the past.

3.2 Parachute and Free-Fall: The 1983-92 Reduction vs. the 2003-07 Reduction

The two troughs in Chart 1 appear very similar. The reduction in Computer Science FTEs from 1983 to 1992 was 31.6%. The decline is even greater when one realizes that 1983 itself was not a maximum compared to the previous years. The 2003-07 reduction was 48.4%. Yet, the loss was far more severe in 2003-07 than in 1983-92.

The reason for this is apparent in Chart 2. Here, not only targeted Computer Science undergraduate enrollment is portrayed, but total departmental enrollment including CS0 and graduate (Master's) courses as well. The 1983-92 trough essentially disappears.

Chart 2: Total Departmental FTEs (Undergrad+Grad+CS0)



What accounts for the smoothing seen at Brooklyn College over the 1983-92 period? Primarily the introduction of the Brooklyn College Core Curriculum that included a Math/Computing course

What accounts for the smoothing seen at Brooklyn College over the 1983-92 period? Primarily the introduction of the Brooklyn College Core Curriculum that included a Math/Computing course (CS0, if you will) that was required of all students in the College. This requirement added students to the department's responsibilities and maintained the overall teaching load of the department. Additonally, the department's graduate enrollment was increasing during this period. More about this, later.

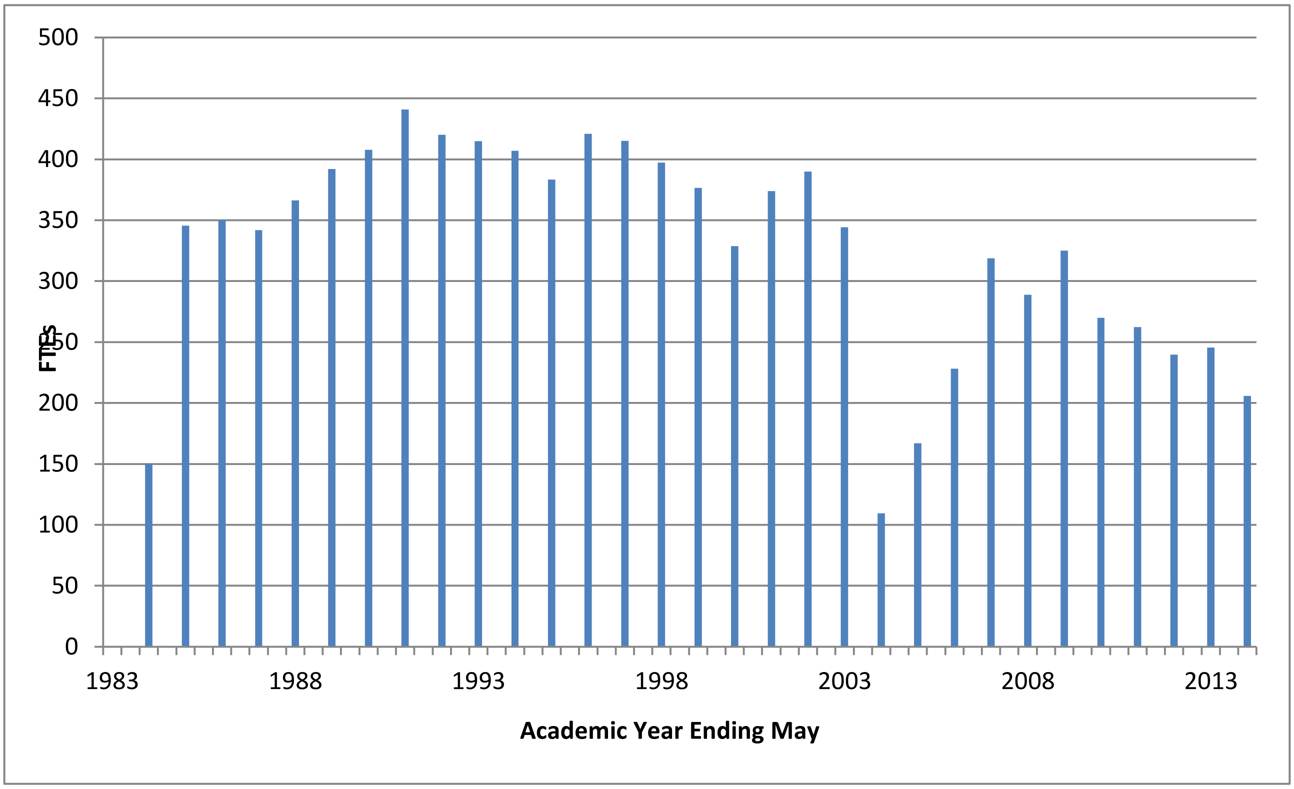
By contrast, the 2003-07 period was a time when more students in the City University were placed initially in community colleges, so a large portion of the College population was transfer students who often transferred in credit for CS0 and no longer took it at the College. As a result, not only did the department suffer the general nationwide disinterest in computer science courses, but also suffered a reduction in enrollment in the "required" computing course, CS0.

4. CS0 Enrollment

Chart 3 shows the department's enrollment in Core computing courses over the 30-year period. The entire Core and the Core computing course(s) underwent significant changes over the 30 years represented by Chart 3. The Core was first introduced in 1980 at the College. The required computing course was actually a combined Math/CS course that taught rudimentary programming in BASIC and mathematics in the form of number theory or probability. Programming was used primarily to illustrate or demonstrate the mathematics. Half the courses were taught by appointees of the Mathematics Department, and half by appointees of the Department of Computer and Information Science. Half the number of students and half the student FTEs were credited to each department.

The CS department chafed over this course, feeling that its computer content was not modern and perhaps that Mathematics was "driving the bus." Professor David Arnow, after much effort, succeeded in partitioning the subject matter of the course into a math version and a computing version. The computing course became a low level mathematical

Chart 3: Core Computing FTE Enrollment



logic and programming logic course using the Pascal language. A text was produced by Arnow and Naomi Bushman [6] in 1995 specifically for this version of the course, and the curriculum and text were used for CS sections of the course through 2003.

There was some dissatisfaction by a number of pedagogues in the College that a single course was used to teach two different subjects, and a number of proposals were put forward. This included leaving the course alone; restoration of the original version of the course, using BASIC, for all sections; and formally creating two different courses. Eventually, two separate courses were formally created beginning in Spring 2003, and students chose which one they wanted. However, certain areas of the College, such as the School of Education, insisted on its students taking the Mathematics version.

The computing course (now designated as Core Curriculum 1312) also morphed into The Nature, Power and Limits of Computing [7], a course in algorithmic thinking, elementary programming, the Internet and the Web. Javascript was used as the language.

For reasons that are unclear, the enrollment in CS0 core courses dropped sharply as soon as this happened, but recovered somewhat in the ensuing years. One external factor in the lower level of Core students across the College was the sharp rise in transfer students who received credit for many Core courses upon transferring in to Brooklyn.

In Fall 2006, the nature of the college-wide Core was changed significantly. The Core was broken up into a lower tier and an upper tier. The lower-tier consisted of 9 required courses (with the same limited choices as before for 3 of the 9). The choice of mathematics or computer science was preserved in the lower tier and the CS0 course we were teaching evolved somewhat, but was not modified significantly at this time.

The upper tier was a wide choice of topical courses, without prerequisites, that departments proposed and retired based on teaching availability, topicality and societal and academic trends. Examples of such courses are Forensic Chemistry; The Silk Road; Oil, Water and Population. Computer Science taught two such courses: Introduction to Robotics, and Paradoxes and the Limits of Knowledge [8] (the latter with the Department of Philosophy and using a text written by a member of the Department). The CS upper-tier courses are among the most popular of the entire upper tier.

Transfer students, who now represent the majority of the college's student population, are often exempt from the lower tier, but still must take two upper-tier courses. Chart 3 includes both lower- and upper-tier Core courses. At the present time, lower tier CS0 enrolls 45 FTEs, while the two upper-tier CS courses together enroll 76.

A new administration at the College, beginning in 2009, sharply reduced the number of adjuncts and other part-time faculty throughout the College. For example, the number of part-time CS instructors dropped from approximately 40 to approximately 15. In addition, the number of full-time CS faculty dropped slowly, from 31 to 26. Class sizes grew from 20-30 to 45 and CS1 was constrained in its enrollment.

4.1 Pathways

As this is being written, the City University, of which Brooklyn College is one constituent, is implementing a University-wide framework, called Pathways, for all general education. The number of gen-ed courses to be required is lower than what Brooklyn College required traditionally. Courses in a number of "course buckets" (such as "Our Scientific World," "Mathematics and Problem Solving," "Creative Expression," "The Individual and Society") were required. The individual colleges of the University had leeway in the courses they could choose to meet the requirements of buckets. A Mathematics and Quantitave Reasoning course was mandated by the University, and Computer Science did not qualify in that course bucket.

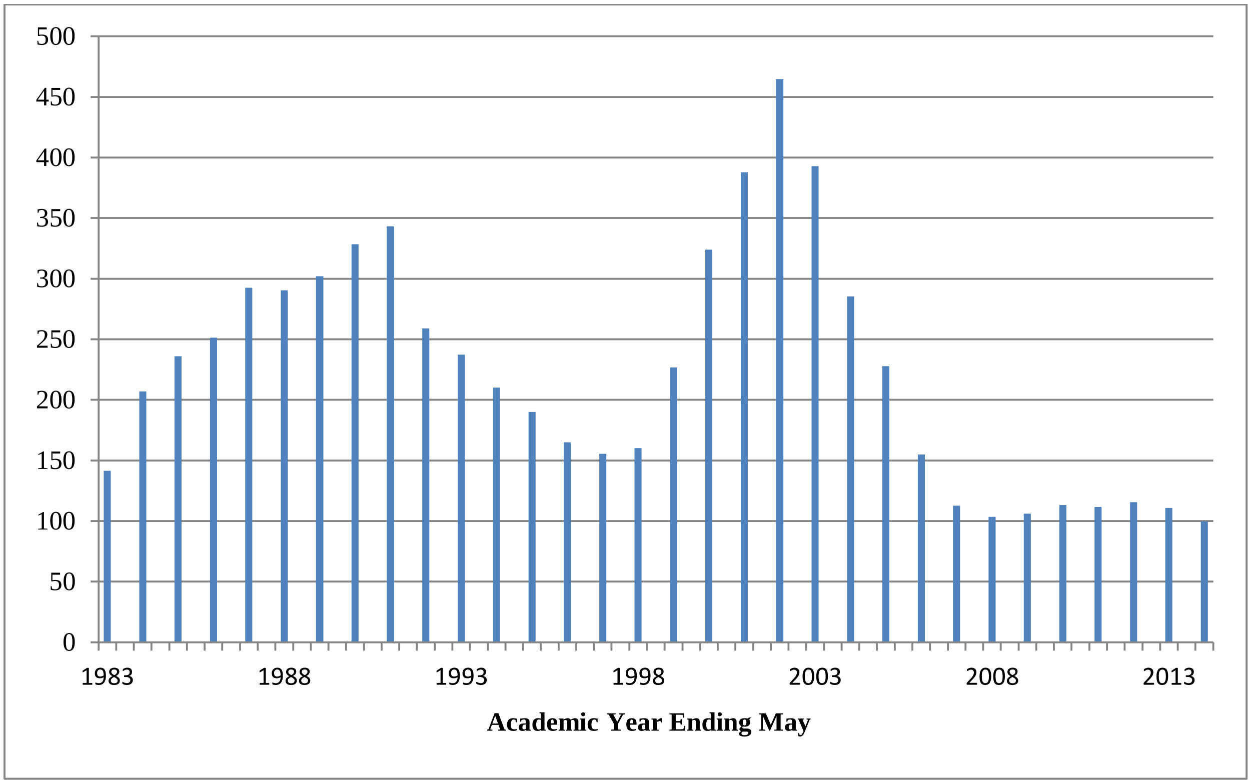
As of now, the Brooklyn College administration has chosen not to include computing in the "Our Scientific World" bucket either, and to omit CS0 entirely as a course meeting the lower-tier general education requirement at the school, despite the fact that the department had prepared four possible CS syllabi and approved three for use as CS0. All upper-tier courses (including the two CS courses), however, will remain as choices for Brooklyn College students in a "college-option" portion of the Pathways framework. The faculty is working on yet another version of general education requirements and it is unknown how Computer Science will fare in the re-organization.

5. Master's Program Enrollment

The Department offers two Master's Programs: in Computer Science and Information Systems. The

Master's in Information Science began in 1987. See Chart 4 for graduate enrollment. The drop from 1991-1997 and the increase from 1998 to 2003 mirrors the change in undergraduate student FTEs, but the graduate enrollment has not recovered since its deep low in 2008, unlike the undergraduate enrollment. Part of the reason for this are Federal administrative rules that make it more difficult for international students to attend, and College and State regulations that make it more difficult to accept students with degrees in other disciplines on the condition that they make up undergraduate foundational work in computer science. Currently, half the Master's students are CS majors and half are IS majors. Applications are now strong.

Chart 4: Master's Program FTE Enrollment



6. First-Level Course Enrollment: CS1 and the Computer Applications Course

An interesting footnote is the comparison between enrollments in CS1, the Introductory Programming course, and the Introductory Computer Applications (CA) course. The latter course teaches word processing, spreadsheet use, a database system and presentation software.

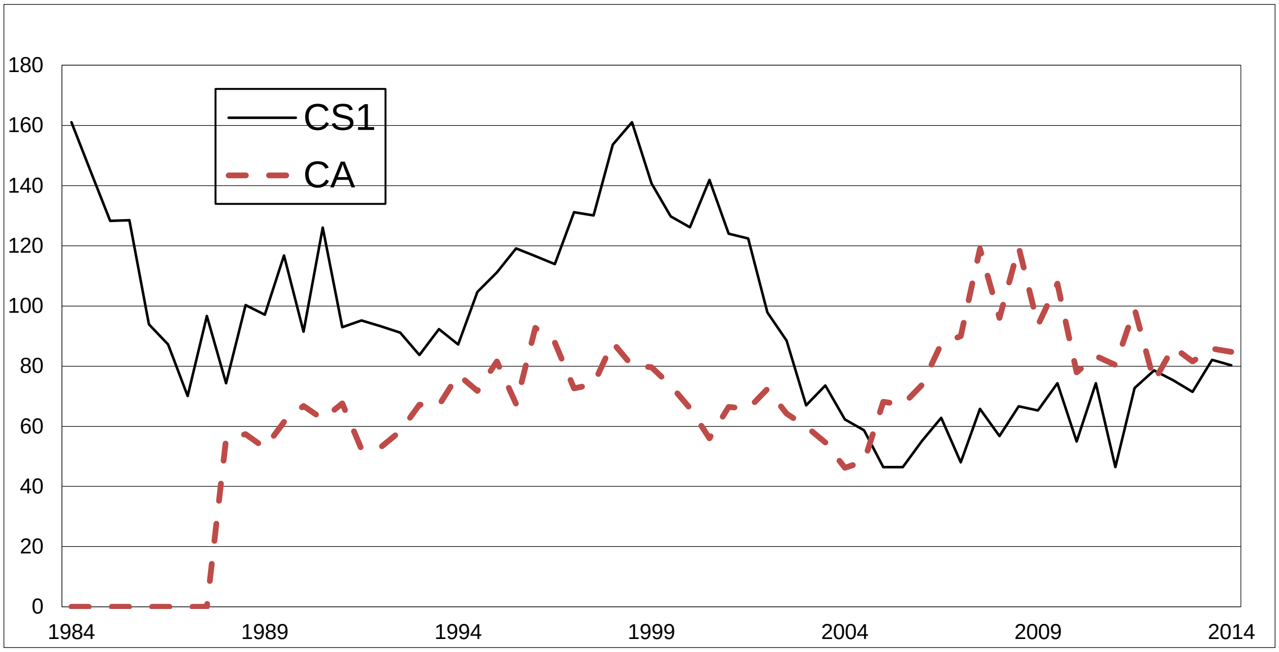
Before the advent of personal computers, the only introduction to computing was CS1, which was recommended by the Business and Accounting programs. Once the CA course was introduced, there was another "introductory computing" course available, seemingly of much more use to Business students. See Chart 5 for the relative enrollments.

We anticipate, however, that CS department enrollment in the Computer Applications course will shrink as more entrants to Business majors come to the College with a prior knowledge of much of the material and as the Business departments incorporate the Applications skills into their own courses.

7. Conclusion

The cyclical nature of computing course enrollments is demonstrated at one urban institution over a period of 31 years, as an example of general trends. The various components of these enrollments show different patterns, often based on local conditions, technological events, perception and realities of the job market, and changes in the nature of courses.

Chart 5: "Introductory Computing" Course Enrollments



References

[1] http://www.cuny.edu/about.html

[2] **Georgetown University Center on Education and the Workforce,** *Best Jobs in America*, Figures 6 and 7 (2012).

[3] Ibid.

[4] **Marketplace,** Why more engineers are losing their jobs, http://www.marketplace.org/topics/business/fallout-financial-crisis/why-more-engineers-are-losing-jobs (January 2014).

[5] **Georgetown** op. cit.

[6] **Arnow, D.** and **Bushman, N.**, *Logic, Circuits and Programs: A Core Approach To Mathematical and Computer Literacy*. Core Press, Brooklyn, NY (1995).

[7] *Computer Science: CORC 1312, Computing: Nature, Power and Limits*, Brooklyn College, Pearson Custom Publishing, 2010, ISBN # 055872820.

[8] ***Yanofsky, N.,*** *The Outer Limits of Reason*, The MIT Press, Cambridge MA (2013).

SESSION

TEACHING METHODS + TEAM TEACHING + TEACHING SUPPORT SYSTEMS AND TOOLS + CASE STUDIES

Chair(s)

TBA

Does Badge-Based Learning Buck the Grading Curve? An Educational Experiment in Computer Architecture

Peter Jamieson (jamiespa@miamioh.edu)
Dept. of Electrical and Computer Engineering
Miami University

Abstract—*In this paper, we look at how a course implemented as a traditional lecture and as a badge-based course compare in terms of grade distribution. Our pilot badge-based course was taught in 2013 and 2014, and our grade results show that the grade distribution was similar to the lecture-based course with the added benefit of students at the upper end of the class demonstrating significantly deeper knowledge in the subject matter. The low-performing students, arguably, learned the same amount of material as they would have in a lecture based course, and benefited from not wasting their own time (and others) in participating in things they were not interested in. Similarly, more ambitious students were able to pursue interest paths within computer architecture at their own pace.*

Keywords: Badges, Grades, Computer Architecture

1. Introduction

In this paper, we look at a computer architecture course implemented as a traditional lecture and as a badge-based course. The key question we have with the non-traditional badge-based course is would the grade distribution be significantly different than a lecture based course? In a worse case scenario, we might find that all students fail the badge-based course or, oppositely, all receive an excellent grade, which does not, normally, happen in our lecture based course taught two times previously.

A badge-based course provides the learner with a set of badges that can be worked on and demonstrated to provide evidence of mastery of a specific set of skills. In our course, badges are distinguished in three categories: basic, intermediate, and advanced. There are three basic badges that must be completed to get a passing grade in the course. Completion of additional badges maps a student's grade to higher and higher letter grades in the course. The motivation behind badge-based learning for this course is to allow people with high interest to delve deeper into the material while still providing less interested students a basic knowledge. Also, badge-based learning is assessed via deliverables and instructor interviews and can be redone if not satisfied. This assessment differs from tests and assignments, which are one off assessment points that can go poorly independent of a students learned skills.

We have taught the lecture based architecture course two times before teaching the course with our badge-base approach in 2010 and 2011. The class size was approximately 40 for the lecture based courses and 10 and 20 for the badge-based course, respectively. The grade distribution was similar for all courses taking into account the small population for the badge-based course. Note, that the results for the 2014 are still in progress. We are excited with the results with this preliminary study since during the badge-based course, we noticed that interested students delved significantly deeper into computer architecture, and less motivated students learned the same material as they would have in a lecture based course.

2. Background

Badge-based learning hinges on ideas of project-based learning [1], experiential learning [2], and formative assessment [3], [4] among others. Recently, there has been renewed attention on badge-based learning partly because of the competitions sponsored by the Mozilla Open Badges Framework [5] and the Digital Media + Competition [6]. Because of these new activities a number of studies ([7], [8], [9], [10]), innovations ([11]), and articles ([12], [13], [14]) have emerged that deal with this learning approach.

The idea of a merit based badge evolved from the British army, and today badges are common among organizations such as girl and boy scouts among many other groups. The base idea of a merit badge is that the badge is awarded once an individual completes a set of tasks, and in an educational setting these tasks demonstrate mastery of a particular topic. The digital badge, which is a new term, is a badge that is awarded and stored on-line as opposed to made, given, and displayed on a uniform. Gibson *et. al.* article [14] provides a good introduction to this emerging educational paradigm.

Even though there has been increased research activity in badge-based learning, there is not a significant body to leverage in this domain. This is the first, to our knowledge, paper that compares the assessment impact of badge-based learning to the traditional grade distribution of a course. We, however, do not adhere to any need for matching grade distributions between the lecture and badge-based versions of the course. Anecdotally, we believe that students in the badge-based learning approach learned much more than those in a traditional lecture setting though we do not attempt to quantify this.

The hope for badge based learning is to change the motivational structure and assessment of mastery, but however, these reward based mechanisms can themselves have problems; a good summary of some of the concerns can be found in Kohn's book "Punished by Rewards" [15]. This paper is not proposing that badge based learning is the only methodology to teach computer architecture and instead is experimenting with one small piece of the puzzle.

3. Course Organization

In this section, we describe the material taught in computer architecture and detail how the two courses are organized. We make some attempt on quantifying how the two courses compare in terms of deliverables.

3.1 Course Learning Outcomes

At Miami University, computer architecture is taught in the second year of a computer science and computer engineering degree and covers basic computer architecture and the associated assembly language. This material aligns well with two common textbooks used to teach undergraduate students computer architecture; Patt and Patel's, "Introduction to Computing Systems - from bits & gates to C & beyond" [16], and Patterson and Hennessy's, "Computer Organization and Design - The Hardware/Software Interface" [17].

Patt and Patel's book is a bottom up approach to learning about computer architecture. The textbook starts by describing the digital components that make up the system, and then continues with more and more complex topics in computer architecture until the student begins to understand how a high-level program, written in the C language [18], is mapped to a processor and executed. Their approach is accompanied with the LC-2 computer simulator and tools that allows students to develop assembly programs and execute them on this machine. The instruction set architecture (ISA) is created for the LC-2 machine with a focus on providing the learner with a comprehensible language, but this ISA is not used in commercial processors.

Patterson and Hennessy's book approaches the same topic by providing a similar bottom approach that can be customized for students depending on if they are hardware or software focused. Their book is accompanied with software for the SPIM simulator [19] which simulates the MIPS [20] ISA. Patterson and Hennessy are also popularly known for a quantitative approach to learning computer architecture in their book, "Computer architecture: a quantitative approach" [21], which is more focused towards students who will actually be working in the computer architecture field.

The prerequisites for the computer architecture course include introduction to programming and data structures, meaning all the students have been exposed to programming languages (normally JAVA programming). Students taking the lecture version of this course come from both computer science and computer engineering. For the badge-based course, there is an additional prerequisite in digital system design, which only computer engineers will have. This allows a the badge-based course to have less focus on number conversions, binary arithmetic, and digital circuits used in the computer architecture.

Some of the outcomes of our computer architecture course are listed here specifying that the student will have the ability to:

1) describe the operations performed by the CPU.
2) enumerate the registers in a CPU and describe their uses.
3) convert unsigned integers between the following representations: decimal, binary, octal, and hexadecimal.
4) represent signed integers using one's and two's complement representations.
5) perform addition and subtraction of signed integers represented in two's complement representation.
6) describe the salient aspects of the stored program concept.
7) describe the key components of a CPU and their functionality.
8) describe commonly used instructions, their formats, operands required, and encoding to opcodes.
9) illustrate the relationship between mnemonics and machine language translation.
10) use selected assembler directives for assembly language programming.

11) describe the various memory addressing modes used by the instructions with example usage.
12) describe the concept of CPI and quantitatively compare performance of various architectural solutions.
13) explain the concept and use of microprogramming
14) describe the process of linking and loading programs.
15) describe the various phases of assembly.
16) develop assembly language programs, debug assembly programs, and trace the operation of assembly language programs.
17) describe the process of interrupt handling.
18) describe the concepts and relationships between physical and virtual memory.
19) describe the four stages in a traditional pipeline
20) describe the taxonomy and categorization of computers into SISD, SIMD, MISD, and MIMD architectures.
21) describe the concept of superscalar processors
22) describe conventional bus and network connections.

From a more general perspective, we might state that once completing this course, students will be able to build simple assembly programs and understand how these programs run on a processor so that they may progress to understanding and performing in courses on Operating Systems, High Performance Computing, and Compilers.

3.2 Lecture Course Organization

Table 1 shows the evaluation structure of the lecture based course. This is a traditional course layout where quizzes and assembly programming assignments are spread out throughout the semester, and exams are used to assess the progress of students. The assembly assignments increase in complexity ending with a task of converting a 'C' program into assembly instructions that includes string manipulation, dynamic memory allocation, and function calls. The assembly language used is MIPS, and a basic architecture is discussed that consists of the four phases of: instruction fetch, instruction decode, execute, and write back. This model can be understood by both the computer scientist and the computer engineer, but some of the finer details such as multiplexing and binary addition are briefly studied to allow the computer scientists to have some understanding of how the computer operates without taking a full course on digital system design.

One observation of this course organization, regardless of right or wrong, is that more than half the grade is allocated to test based assessment (60%). The other half of the course evaluation is split between programming and in class activities (some based on our previous work [22]). Because of this structure, a lot of time is spent in administering and preparing for tests.

3.3 Badge-based Course Organization

The badge-based course consists of a student demonstrating their increasing understanding of computer architecture by completing badges. The structure of these badges is shown in Figure 1 where three core or basic badges are at the top of the diagram, intermediate badges are under them, and advanced badges are along the base. In this figure, solid lines connecting badges are prerequisites for higher level badges, and the dotted lines are used for related badges (as in you should be familiar with one of the dotted lines before pursuing the targeted badge).

Each of the badge details is maintained on a course WIKI, and Figure 2 shows a sample badge for the "assembler basic badge". Note that there is a circular badge with a red border in the upper left corner, and this is the digital badge that we created and is displayed on a students wiki page to show what they have achieved. Once a student has prepared the badge, they will bring the badge deliverables to the instructor and a discussion will occur to verify that the material is sufficiently mastered and understood. If the student has mastered the badge material, then the badge will be added to the students set of completed badges such as in Figure 3, which is an example of a students badges at the completion of the course in 2013 (3 core badges, Intermediate System Skills, Intermediate Assembly, and Advanced Compiler).

Class time in a course such as this, is spent working on badges with the added bonus that the instructor is available to answer questions. The instructor also evaluates each badge during class time or during office hours. The remaining logistical question is how are the badges related to course grades. To achieve this we use a mapping from badges to grades that is provided at the beginning of the semester in the syllabus. Table 2 shows how the badges are mapped to letter grades, and this is done for administration and economical purposes of the university. In terms of an educational benefit, the authors do not believe grading has any benefit in the learning process.

Table 1
LECTURE COURSE EVALUATION AND STRUCTURE

Item	Percentage of Course Grade	Additional Details and Topics Covered
Exam I	15%	1-11
Exam II	15%	1-18
Exam III	20%	All Topics
Class Participation	15%	Includes attendance and in-class execution of programs [22]
Quizzes	10%	Each quiz worth 1%
Assignment I Assignment II Assignment III Assignment IV	6% 6% 6% 7%	MIPS Assembler programs built on MARs Emulator

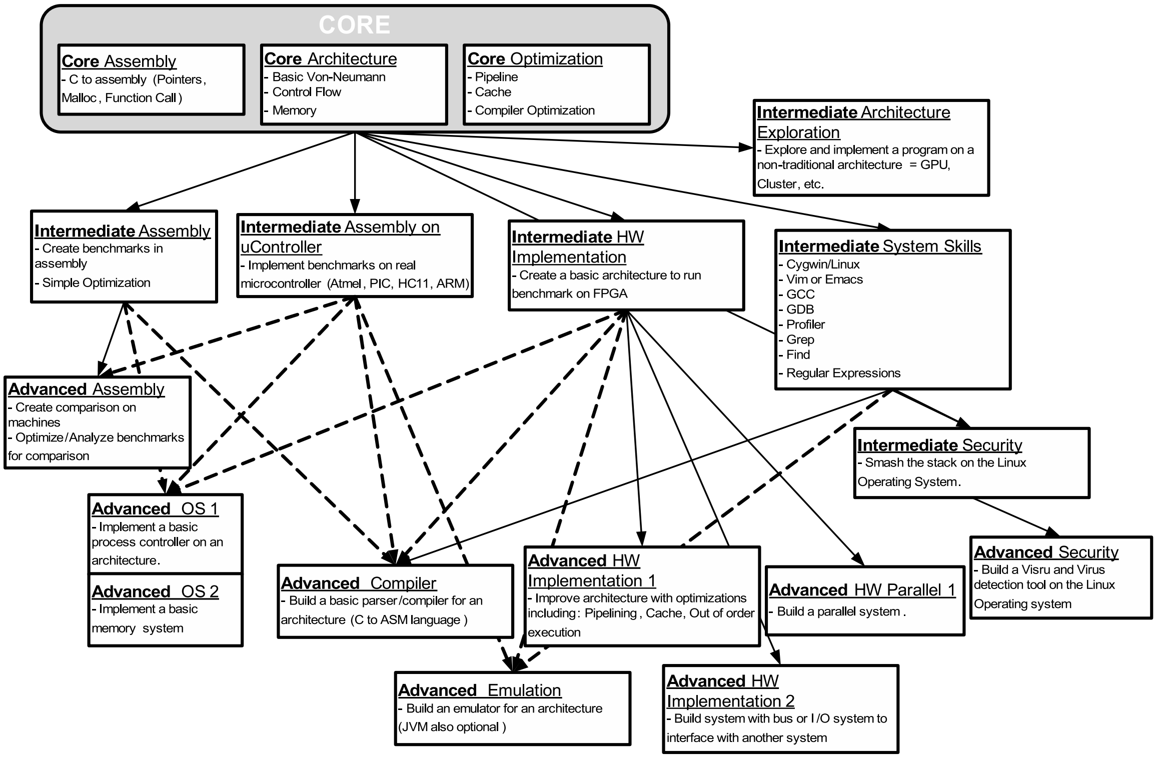

CORE
Core Assembly
- C to assembly (Pointers, Malloc, Function Call)
Core Architecture
- Basic Von-Neumann
- Control Flow
- Memory
Core Optimization
- Pipeline
- Cache
- Compiler Optimization
Intermediate Architecture Exploration
- Explore and implement a program on a non-traditional architecture = GPU, Cluster, etc.
Intermediate Assembly
- Create benchmarks in assembly
- Simple Optimization
Intermediate Assembly on uController
- Implement benchmarks on real microcontroller (Atmel, PIC, HC11, ARM)
Intermediate HW Implementation
- Create a basic architecture to run benchmark on FPGA
Intermediate System Skills
- Cygwin/Linux
- Vim or Emacs
- GCC
- GDB
- Profiler
- Grep
- Find
- Regular Expressions
Advanced Assembly
- Create comparison on machines
- Optimize/Analyze benchmarks for comparison
Intermediate Security
- Smash the stack on the Linux Operating System.
Advanced OS 1
- Implement a basic process controller on an architecture.
Advanced OS 2
- Implement a basic memory system
Advanced Compiler
- Build a basic parser/compiler for an architecture (C to ASM language)
Advanced HW Implementation 1
- Improve architecture with optimizations including: Pipelining, Cache, Out of order execution
Advanced HW Parallel 1
- Build a parallel system.
Advanced Security
- Build a Visru and Virus detection tool on the Linux Operating system
Advanced Emulation
- Build an emulator for an architecture (JVM also optional)
Advanced HW Implementation 2
- Build system with bus or I/O system to interface with another system

Fig. 1
THE PRE AND CO-REQUISITES FOR THE BADGES IN COMPUTER ORGANIZATION.

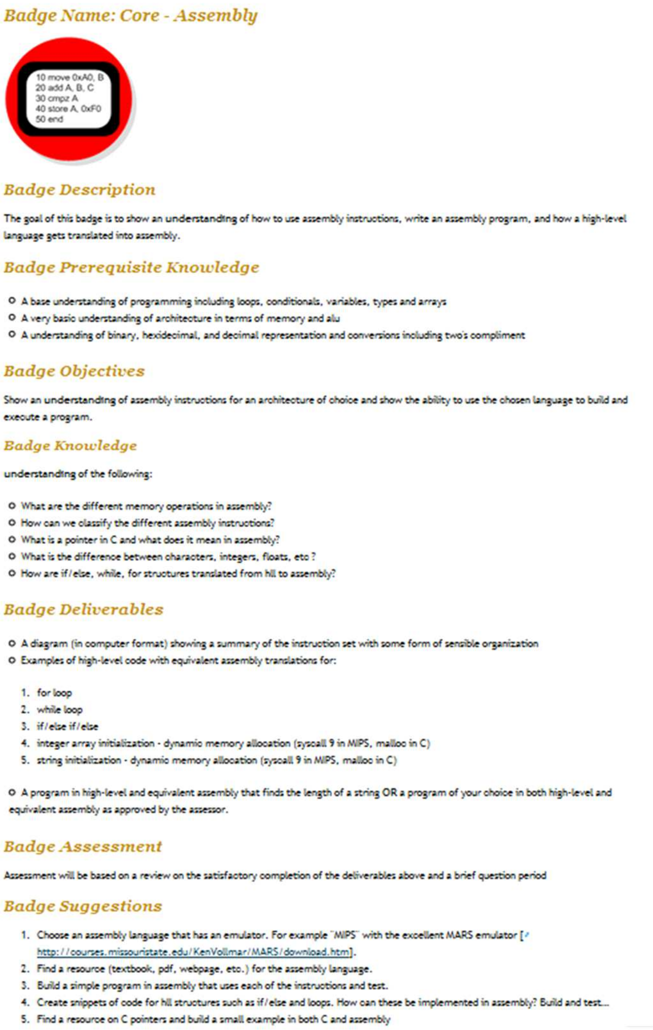
4. Results

The question we had for this course is not if the students would learn more or less in a badge-based course format (and we believe that students learn significantly more based alone on the topics and deliverables they produced). Instead, we were concerned that a mastery driven second year course would be biased all towards failure or all towards success as shown in the grade distribution. To observe this we compare the grades in the courses. Note that we currently do not have grades

Table 2

A TABLE SHOWING THE MAPPING BETWEEN BADGES AND LETTER GRADES.

Letter Grade	Core Badges	Intermediate Badges	Advanced Badges	Additional Details
F	0	-	-	-
D-	0-3	-	-	Base badges partially assessed
D	1	-	-	-
C	3	-	-	-
B	3	1	-	-
B+	3	2	-	-
A-	3	3	-	-
A	3	1	1	-
A+	3	1	1	Exceptional advanced work



Badge Name: Core - Assembly

Badge Description

The goal of this badge is to show an understanding of how to use assembly instructions, write an assembly program, and how a high-level language gets translated into assembly.

Badge Prerequisite Knowledge

- A base understanding of programming including loops, conditionals, variables, types and arrays
- A very basic understanding of architecture in terms of memory and alu
- A understanding of binary, hexidecimal, and decimal representation and conversions including two's compliment

Badge Objectives

Show an understanding of assembly instructions for an architecture of choice and show the ability to use the chosen language to build and execute a program.

Badge Knowledge

understanding of the following:

- What are the different memory operations in assembly?
- How can we classify the different assembly instructions?
- What is a pointer in C and what does it mean in assembly?
- What is the difference between characters, integers, floats, etc ?
- How are if/else, while, for structures translated from hll to assembly?

Badge Deliverables

- A diagram (in computer format) showing a summary of the instruction set with some form of sensible organization
- Examples of high-level code with equivalent assembly translations for:

  1. for loop
  2. while loop
  3. if/else if/else
  4. integer array initialization - dynamic memory allocation (syscall 9 in MIPS, malloc in C)
  5. string initialization - dynamic memory allocation (syscall 9 in MIPS, malloc in C)

- A program in high-level and equivalent assembly that finds the length of a string OR a program of your choice in both high-level and equivalent assembly as approved by the assessor.

Badge Assessment

Assessment will be based on a review on the satisfactory completion of the deliverables above and a brief question period

Badge Suggestions

1. Choose an assembly language that has an emulator. For example "MIPS" with the excellent MARS emulator [http://courses.missouristate.edu/KenVollmar/MARS/download.htm].
2. Find a resource (textbook, pdf, webpage, etc.) for the assembly language.
3. Build a simple program in assembly that uses each of the instructions and test.
4. Create snippets of code for hll structures such as if/else and loops. How can these be implemented in assembly? Build and test...
5. Find a resource on C pointers and build a small example in both C and assembly

Fig. 2

SAMPLE SCREEN SHOT OF ONE OF THE BADGES.

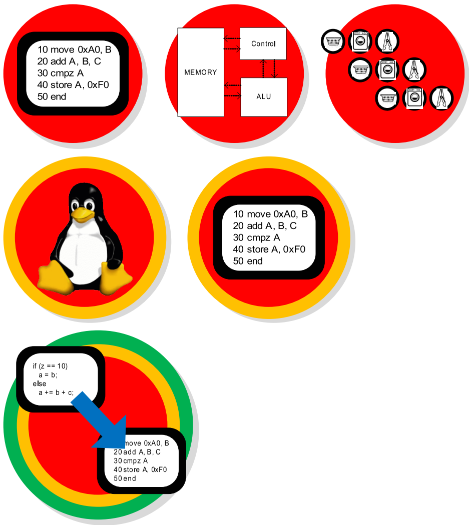


Fig. 3

A SAMPLE OF BADGES COMPLETED BY ONE STUDENT IN 2013.

for the 2014 class, but these will be included in our final camera ready version.

Figure 4 show the course grade distribution for each of the years the course was taught (2013 and 2014 the course was taught in the badge-based form). The sample size is significantly different when comparing 2010 and 2011 to 2013 and 2014 because the badge-based course was taught to computer engineers only and not the combination of computer engineers and computer scientists. This difference in students could impact the quality of there grades, but this factor was not studied. Also, no significant conclusions should be made from this data as the sample size is very small for the 2013 and 2014 classes. Finally, these results are for classes

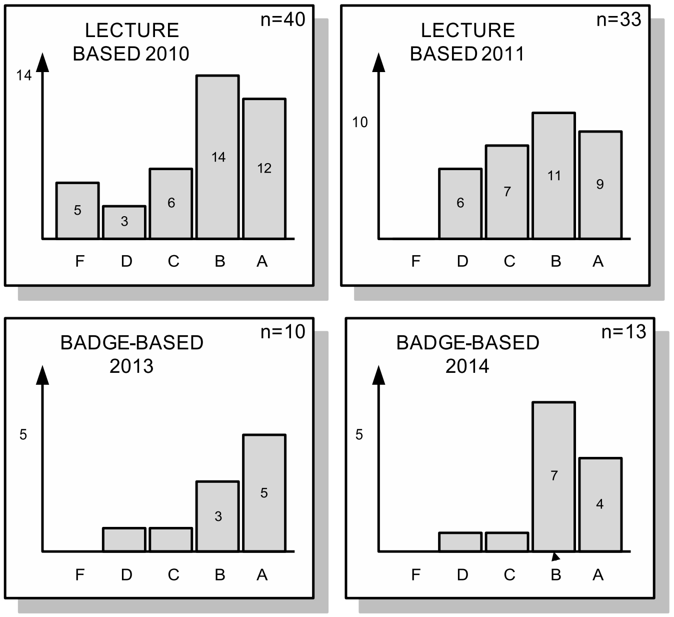


Fig. 4
THE FOUR GRADE DISTRIBUTIONS FOR 2 LECTURE-BASED AND 2 BADGE BASED RESULTS OF THE COURSE.

taught by the same instructor.

The most significant difference between the grades for each of the classes is that there are more A grades compared to B grades for the badge-based course when compared to the lecture-based course. We explain this to the fact that grades in the badge-based course are a long term decision made by the student as opposed to smaller decisions that might result in poor performance on a test or assignment. This is one of the key benefits to a mastery based approach since the student is under no pressure (other than the time in the semester) to perform at very specific assessment times.

An interesting point in terms of a student's choice is that a small percentage of the 2013 and 2014 class selected to just learn the minimal amount of material to get a passing grade. In the badge-based course, this choice had no impact on the other students as they could continue at their own pace on the material that they had chosen to explore deeper. Otherwise, there are not significant differences between the grade distributions. This suggests that a badge-based approach when mapped to grades will show a similar grade distribution when compared to a lecture based course.

5. Discussion

Our results show that the badge-based course has little impact on grade distributions, which from our perspective is good. If our results were biased towards all high letter grades, we, similarly, would be fine as we look to grades as an indication of mastery of the material and not as a categorizing mechanic for which students are relatively rated against one another. The exciting part of this method is the actual student experience, which is independent of much of the administration and organizational data that are described in this paper.

We start this discussion by comparing the core badges to material covered in the lecture-based course. The main difference between these materials is not the content, but how much of the course is spent learning these materials; in the badge-based course these materials are normally completed by the sixth week of study while in the lecture-based course this takes almost the entire semester. Note that the assessment choice, exams and quizzes, take up almost one quarter of the lecture based course in doing and preparing for the exams. We are not claiming that this time is wasted, but in the badge-based course this time is used by the student to learn other topics and to delve into an area of the material that they are most interested in. When considering that these students are training to become professionally practicing engineers explaining deliverables is much more career related than taking one-off tests.

One question the reader might asks is, how does the depth of the material covered in the core badges compare to the entire lecture course? We do not have any conclusive evidence to answer this question, but anectodally, we believe that the badge-based student learners gain almost the same understanding with less assembler programming skill that is regularly practiced in the lecture-based course. However, this low-level programming practice has little industrial impact, in most student's future careers, and spending that design practice on implementing hardware in register transfer level (RTL), assembly programming on microprocessors, learning about architecture and security, or even building and comparing assembly programs are all valuable activities that will help a student better understand computer architecture.

The additional difference with the badge-based course is the advanced badges. These badges allow a student to make significant strides into building and understanding complex pieces of a modern computer. This can be similarly achieved in a lecture based course with a summative project, but the logistic challenges to allow for the wide range of advanced badges to be tackled by all the students causes challenges with this approach. Many computer architecture and organization course with final projects normally focus on advanced assembly programming assignments. The badge-based approach allows students to select the piece of the system that they think they are most interested in.

One final discussion point is the scalability of the badge-based approach. From our perspective, the goal is not scalability, but the concept is still interesting to address and is of interest to the economics of engineering education. The badge-based course currently allows for the instructor to divide the time in the class adequately with all students such that for almost half of the class time there is no student demand. Based, on these rough numbers, we believe that we could hold approximately 25 students without any major reorganization steps. We do not believe that a class size of much greater than 25 can accommodate the students sufficiently without additional teaching assistants or instructors, however, this course could be grown with teaching assistants.

6. Conclusions

In this study, we investigated if there would be significant grade differences between a badge-based learning course and a traditional lecture-based course on computer architecture. I, preliminary, findings show that the grade distributions are very similar noting that the badge-based course was taught only to computer engineers (not computer scientists) and the sample sizes of the two approaches are significantly different. We, also, provided a framework for badge-based courses and discussed what benefits this approach has for students and instructors.

We believe that the badge-based approach to teaching computer architecture has made a significant improvement to what the students are learning. This is the case since badge-based learning allows interested learners to more deeply pursue aspects of computer architecture that they are interested in, which is not really possible in the lecture based approach.

References

[1] J. Macias-Guarasa, J. Montero, R. San-Segundo, A. Araujo, and O. Nieto-Taladriz, "A project-based learning approach to design electronic systems curricula," *Education, IEEE Transactions on*, vol. 49, no. 3, pp. 389 –397, 2006. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1668283

[2] D. A. Kolb *et al.*, *Experiential learning: Experience as the source of learning and development*. Prentice-Hall Englewood Cliffs, NJ, 1984, vol. 1.

[3] R. J. Mislevy and M. Riconscente, "Evidence-centered assessment design," *Handbook of test development*, pp. 61–90, 2006.

[4] M. Heritage, "What do teachers need to know and do," *Phi Delta Kappan*, vol. 89, no. 2, pp. 140–145, 2007.

[5] Mozilla, "https://wiki.mozilla.org/Badges/About," 2014.

[6] D. M. . L. Competition, "http://dmlcompetition.net/," 2014.

[7] S. Abramovich, C. Schunn, and R. M. Higashi, "Are badges useful in education?: it depends upon the type of badge and expertise of learner," *Educational Technology Research and Development*, vol. 61, pp. 217–232, 2013.

[8] F. Khaddage, R. Baker, and G. Knezek, "If not now! when? a mobile badge reward system for k-12 teachers," in *Society for Information Technology & Teacher Education International Conference*, vol. 2012, no. 1, 2012, pp. 2900–2905. [Online]. Available: http://www.editlib.org/p/40029

[9] J. L. Santos, S. Charleer, G. Parra, J. Klerkx, E. Duval, and K. Verbert, "Evaluating the use of open badges in an open learning environment," in *Scaling up Learning for Sustained Impact*. Springer, 2013, pp. 314–327.

[10] A. Bruckman, E. Edwards, J. Elliott, and C. Jensen, "Uneven achievement in a constructionist learning environment," in *International Conference of the Learning Sciences: Facing the Challenges of Complex Real-world Settings*, vol. 7, no. 17. Psychology Press, 2013, p. 157.

[11] S. Charleer, J. Klerkx, J. L. Santos, and E. Duval, "Improving awareness and reflection through collaborative, interactive visualizations of badges," in *3rd Workshop on Awareness and Reflection in Technology-Enhanced Learning*, 2013. [Online]. Available: http://ceur-ws.org/Vol-1103/paper5.pdf

[12] J. Young, "Badges earned online pose challenges to traditional college diplomas," *The Chronicle of Higher Education*, 2012.

[13] J. Antin and E. F. Churchill, "Badges in social media: A social psychological perspective," in *CHI 2011 Gamification Workshop Proceedings (Vancouver, BC, Canada, 2011)*, 2011.

[14] D. Gibson, N. Ostashewski, K. Flintoff, S. Grant, and E. Knight, "Digital badges in education," *Education and Information Technologies*, pp. 1–8, 2013.

[15] A. Kohn, *Punished by rewards: The trouble with gold stars, incentive plans, A's, praise, and other bribes*. Houghton Mifflin Harcourt, 1999.

[16] Y. N. Patt and S. J. Patel, *Introduction to Computing Systems: From Bits & Gates to C & Beyond*. New York, NY, USA: McGraw-Hill, Inc., 2004.

[17] D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/software Interface*. Morgan Kaufmann, 2005.

[18] B. W. Kernighan and D. M. Ritchie, *The C programming language*. Prentice-Hall, 1978.

[19] J. Larus, "SPIM S20: A MIPS R2000 Simulatora," Tech. Rep., 1990.

[20] J. Hennessy, N. Jouppi, S. Przybylski, C. Rowen, T. Gross, F. Baskett, and J. Gill, "Mips: A microprocessor architecture," in *MICRO 15: Proceedings of the 15th annual workshop on Microprogramming*. Piscataway, NJ, USA: IEEE Press, 1982, pp. 17–22.

[21] J. L. Hennessy and D. A. Patterson, *Computer architecture (2nd ed.): a quantitative approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[22] P. Jamieson, D. Davis, and B. Spangler, "The mythical creature approach-a simulation alternative to building computer architectures." in *FECS*, 2010, pp. 23–28.

Transformations in Computer Graphics and Alternative Mathematical Models

Hassan Farhat
Computer Science
University of Nebraska at Omaha

Abstract- *Mathematical transformations are essential component of studying computer graphics. Discussions on transformations are found throughout the graphics pipeline. Transformations help in describing objects in many different coordinate systems. This includes modeling, world, viewing, and display coordinates.*

The discussion of transformations in several of the existing computer graphics textbooks is found dispersed with minimal discussion of relation among these transformations. In this paper, we look at the different transformations used and provide a unified approach of discussing important transformation properties. Of importance are linear and affine transformations. We provide a pedagogy and presentation that helps in a unified coverage of the transformations and their corresponding properties.

The unified discussion includes the need for the two types of transformations. It also includes simple presentation of closure properties under each transformation and how it can easily be verified.

(KEYWORDS: Computer Graphics, Affine Transformations, Linear Transformations, Bases, Orthogonal)

1 Introduction

Transformations are fundamental to computer graphics. They are found throughout the rendering pipeline. They are also found in the hardware realization of GPUs (Graphics Processing Units). In earlier work [4] we presented mathematical foundation rooted in vector spaces.

Figure 1 shows where transformations are needed in a simplified OpenGL graphics pipeline [4]. Vector spaces lend to linear transformations where the transformations work on vectors and scalars only. While vector spaces are based on vectors in the classical understanding, they are Cartesian coordinates origin independent. Many vector spaces are defined over different domains (polynomial domain for example). Linear transformations on vector spaces have the property that the transformation of a special zero vector $\mathbf{0}_v$ in the domain, V, is a special zero vector, $\mathbf{0}_w$, in the range W. Linear transformations on vector spaces account for two important primitive transformations, scaling and rotation. The need to account for additional transformations, most importantly translation, and to include a reference point as an origin of Cartesian system of coordinates, lead to expanding linear transformations to include affine transformations.

The need to include distances and vector/line length lead to inclusion of Euclidian spaces where dot products and vector length can be used. We attempt to unify the discussion of linear and affine transformations and corresponding mathematical properties in use in computer graphics. This includes the discussion of closure properties such as parallelism, ratios, and linearity. We present a discussion that relates both transformations and present discussion of closure properties at the transformation level. Computer graphics textbooks that deal with linear transformations are found in [2] through [3] and [5] through [12].

We formalize the major concepts of transformations using vector spaces, linear transformations on vector spaces, and affine transformations on scalars, points, and vectors. We discuss which properties are preserved under the different transformations. We also look at under what conditions, special cases, other properties are preserved.

The paper is organized as follows. In section 2, we look at the different transformations found in the graphics pipeline. In section 3 we discuss closure properties under transformations. In section 4, we look at the need for a unified representation to encompass both transformations. The conclusion is given in section 5.

2 The Different Transformations

Transformations are applied to objects within the same coordinate system or transform (place and modify) objects relative to another coordinate system. In Figure 1, an object in modeling coordinates is transformed to world coordinates, [4]. In the world coordinate, a synthetic camera (viewing) coordinate system is defined. The object in world is then placed in the viewing coordinate system. The transformations applied during

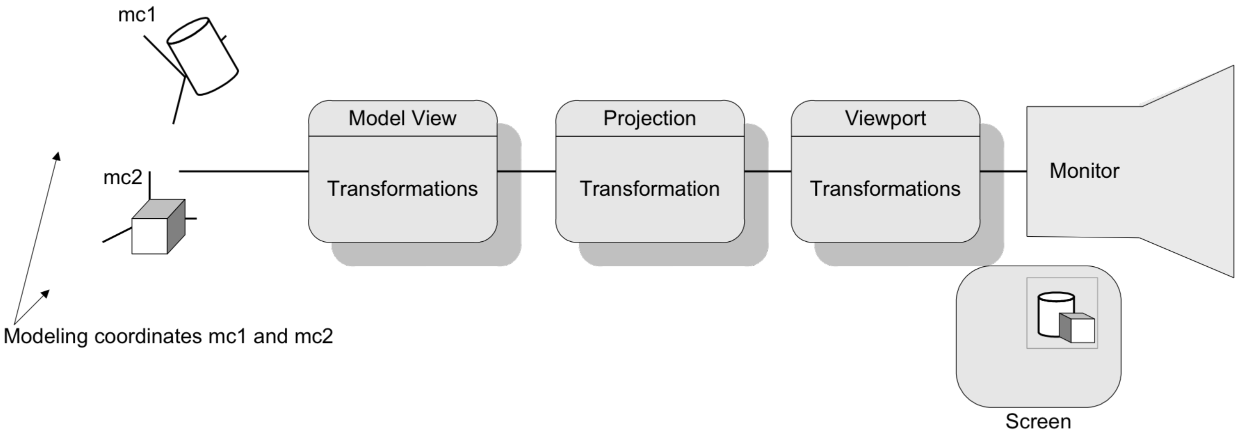


Figure 1: Simplified graphics pipeline

this phase of a graphics pipeline are either linear or affine transformations[1].

Following this phase, clipping volumes described in viewing coordinates are used to clip parts of the scene now described in viewing. The clipping volumes are either a parallelepiped or a frustum. Before clipping is applied, the viewing volumes are transformed to normalized clipping cube where the object along with the view volume is transformed. The view volume is transformed to a cube with center at (0, 0, 0) and with unit length along the positive and negative x-axis, y-axis and z-axis.

After clipping is applied, the final phase is to display the object in screen coordinates. Here the object is first placed in display coordinates with depth information used for rendering.

We look at the first phase of transformations where linear and affine transformations are applied. A linear transformation, L, is a function applied to vector spaces where the domain, V, and the range, W, are vector spaces. A linear transformation is a special function satisfying the following:

(a) For any two vectors $X \in V$ and $Y \in V$ we have $\mathbf{L}(X + Y) = \mathbf{L}(X) + \mathbf{L}(Y)$

(b) For any vector $X \in V$ and any scalar $c \in R$ we have $\mathbf{L}(c.X) = c.\mathbf{L}(X)$

In the definition above, the "+" and the "." are the vector addition and scalar multiplication operators found in the definitions of vector spaces[2]. The symbols on the left of the equality in (a) and (b) represent the operators as defined in the domain vector space V while the symbols on the right represent the operators in the range W.

Of interest in computer graphics, is the case where the vector spaces V and W are such that $V = R^n$ and $W = R^m$ where R^n and R^m represent the set of n-dimensional and m-dimensional vectors sets. The vector addition and scalar multiplication for the domain and range are the normal n-dimensional or m-dimensional vector addition and scalar multiplication.

For this special case one can show that the function, $\mathbf{L}$: $R^n \rightarrow R^m$ where $\mathbf{L}$ is defined as

$$\mathbf{L}(X) = AX \qquad (1)$$

with A an m × n constant matrix, forms a linear transformation.

Using Matrix algebra, the verification that $\mathbf{L}$ forms a linear transformation is straightforward. For any two vectors X1 and X2 we have $\mathbf{L}(X1 + X2) = A(X1 + X2) = AX1 + AX2 = \mathbf{L}(X1) + \mathbf{L}(X2)$. This shows property (a) of the definition is satisfied. For any scalar, $k \in R$, vector $X1 \in V$, we have $\mathbf{L}(kX1) = A(kX1) = kAX1 = k\mathbf{L}(X1)$. This shows property (b) of the definition is satisfied.

How does the above help in computer graphics? a) It provides uniformity to representations of transformations found in the first phase of the computer graphics pipeline. And b) properties preserved under linear transformations can then hold true in computer graphics with the transformation as given in equation (1).

Example applications in computer graphics is in writing a vector as a linear combination of vectors in different bases and hence different coordinate system as shown in Figure 2. The components of the vector P can be written in two different coordinate systems, [4].

The figure shows two bases for the R^2 space, the natural basis set {i, j} and the set {X1, X2}. The position vector

[1] We expand on above in next sections.

[2] The scalar multiplication "." can be implied and dropped

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

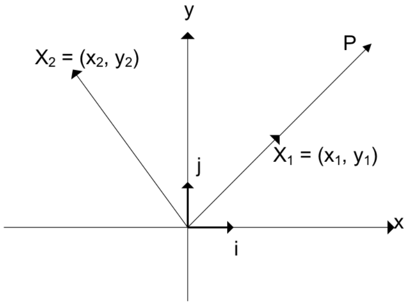


Figure 2: Two coordinate systems; system 1 basis {i, j}; system 2 basis {X1, X2}

components for vector P can be written in two different coordinate systems[3].

$$P = (x, y) = x\text{ i} + y\text{ j} = x\ (1, 0) + y\ (0, 1)$$

The x and y constants form the coordinates of P with respect to the orthonormal basis {i, j}. Similarly, one can obtain the coordinates of P in the different coordinate system composed of the two linearly independent vectors but not necessarily orthogonal vectors X1 and X2 forming the basis {X1, X2}. By inspection we have[4]

$$P = (x, y) = \text{x i} + \text{y j} = 2\text{ X1} + 0\text{ X2}$$

The coordinates of P with respect to basis {X1, X2} is (2, 0).

Other examples applications in computer graphics include the three primitive transformations: reflection about the y-axis, uniform scaling by s, and rotation by an angle θ. The matrix A for each transformation is given as two dimensional matrix: A = scale(s), A = reflect(y) and A = rotate(θ) where

$$scale(s) = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \qquad reflect(y) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$rotate(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Missing from the above is the important translation transformation. Since translation leaves the direction of a vector as well as its magnitude unchanged, and since vectors are position independent, the vectors are unchanged under the translation transformation. As a result, the A matrix in (1) is the identity matrix.

While the translation transformation does not affect vectors, it affects point's coordinate's descriptions. The domain of elements in vector spaces are vectors. The need to include points in the domain leads to introducing the students to the fundamental concepts of affine transformations.

An affine transformation, **τ**, can be thought of as a generalized form of linear transformations. Over the two sets V and W, we have

$$\tau(X) = AX + B \qquad (2)$$

where B is a constant element from W, and A is a matrix as given in (1).

The above equation can be applied to points as well as vectors, and hence the space of V and W are no longer just vectors. In affine space, point subtraction is defined to yield the traditional vector result. Hence in affine spaces, V and W are closed under point subtraction. From point subtraction, points can be written as a sum of a point and a vector. Unlike vectors, however, the sum of two points in affine spaces is undefined in general. Instead a special sum of points, affine sum, is defined. Euclidean spaces are affine spaces with the concept of inner product included. This results in geometric representations of the traditional Cartesian coordinates system with the concepts of distances included. We consider these concepts in teaching computer graphics next.

3 The Closure Properties

We discuss closure properties of affine and linear transformations. These properties greatly simplify the task of rendering. We present sample discussion of closure properties that is based on matrix algebra. Alternative discussion is presented in [9]. The advantage of the presentation is in the simplicity of applying the discussion to arbitrary dimensions.

The basic concept needed is the parametric equation of a line. We show the students how the equation is generated from the definition of parallel vectors and the definition of vector generation from point subtraction in affine spaces as follows.

Two vectors V1 and V2 are said to be parallel if there exists a constant k such that V1 = kV2. For two given points P_0 and P_1, the equation of the vector $\overrightarrow{P_0P_1}$ is given as

$$\overrightarrow{P_0P_1} = P_1 - P_0 = V \qquad (3)$$

$$P_1 = P_0 + V \qquad (4)$$

Equation 4 is derived from equation (3) and is known as a point vector addition. As can be seen the sum yields a point. Given an arbitrary point, Pi and an arbitrary vector

[3] The figure is for illustration; while the figure shows an origin "point", vector spaces have zero vectors but no points in there domains.
[4] The components of the vectors X1 and X2 are given in basis {i, j}.

V, then a point P(u) on the line passing through Pi and parallel to V can be written as

$$\overrightarrow{P_iP_{(u)}} = P_{(u)} - P_i = uV \tag{5}$$

where u is a scalar; V is called a direction vector. When the points given are P_0 and P_1, equation 5 can be rewritten as

$$\overrightarrow{P_0P_{(u)}} = P_{(u)} - P_0 = u.(P_1 - P_0) \tag{6}$$

Equation (6) can be rewritten as

$$\begin{aligned} P_{(u)} &= P_0 + u(P_1 - P_0) \\ &= (1-u)P_0 + uP_1 \end{aligned} \tag{7}$$

In equation (7) the arbitrary point $P_{(u)}$ is written as a sum of two points. This is the special case where the sum of two points is defined; it is called affine sum. For the sum to be defined, the sum of scalars multiplying the points must add to 1. The above equations serve in the verification of many affine transformations properties. We introduce the students to these properties. We illustrate sample examples.

In the verification we use matrix properties over matrix addition and multiplication with scalars included. We start with the property that affine transformations are closed under lines; lines are transformed to lines under affine transformations. Hence it is enough to transform the endpoints of a line through the transformation pipeline. The line is reconstructed from the transformed endpoints in the last stages of the rendering pipeline.

Let P_0 and P_1 represent the endpoints of a line segment. Hence, $\tau(P_0)$ and $\tau(P_1)$ are the transformed points under the affine transformation. Our task is to show that an arbitrary point on the line P_0P_1 is transformed to an arbitrary point on the line formed by the transformed points.

Let the arbitrary point be P(u). When τ is applied to the three points we have

$$\begin{aligned} &\tau(P_0) = AP_0 + B, \quad \tau(P_1) = AP_1 + B, \\ &\tau(P_{(u)}) = AP_{(u)} + B \end{aligned} \tag{8}$$

From equation (7), we have[5]

$$\begin{aligned} &\tau(P_u) = \tau((1-u)P_1 + uP_2) \\ &= A((1-u)P_1 + uP_2) + B \\ &= (1-u)AP_1 + uAP_2 + B \\ &= (1-u)AP_1 + uAP_2 + uB + (1-u)B \\ &= (1-u)\tau(P_1) + u\tau(P_2) \end{aligned} \tag{9}$$

Equation (9) shows the verified result, the point $P_{(u)}$ on the original line is transformed to a point on the transformed line. In fact equation (9) includes another important result. The parameter, u, represents the fraction of the distance of the point P(u) along the directed line with initial point P_0 and final point P_1; u = 0, 0.1, 0.5, 1 represent, respectively the point P_0, the point at 1/10 the distance between P_0 and P_1, the point at 1/2 the distance between P_0 and P_1 and the point P_1. As seen in equation (9) the affine transformation preserves this fractional distance (same u value is used).

With the above affine (parametric) equations other significant affine transformation properties can be verified in similar simple fashion. Examples include preserving parallelism and intersection of lines. For parallelism we make use of the fact that two vectors V_1 and V_2 are parallel if there exists a non-zero scalar k such that $V_2 = kV_1$.

Assume given two line segments, L1 and L2, with P_0, P_1 the endpoints for L1, and P_2, P_3 the endpoints for L2. If line L1 is parallel to line L2 then

$$\begin{aligned} P_{L1(u)} &= P_0 + u(P_1 - P_0) \\ P_{L2(u)} &= P_2 + u(k(P_1 - P_0)) \end{aligned} \tag{10}$$

For a scalar k, k ≠ 0, equation (10) represents the parametric equation of two parallel lines since the direction vectors are scalar multiple of each other (V and kV). To show the transformed lines are parallel, it is sufficient to show the direction vectors of the transformed line are scalar multiple of each other. From (10), we have

$$\begin{aligned} \tau(P_{L1(u)}) &= A(P_0 + u(P_1 - P_0)) + B \\ &= \tau(P_0) + uA(P_1 - P_0) \\ \tau(P_{L2(u)}) &= A(P_2 + u(k((P_1 - P_0))) + B \\ &= \tau(P_2) + u(kA(P_1 - P_0)) \end{aligned} \tag{11}$$

The transformed lines are parallel (the direction vectors are parallel $A(P_1 - P_0)$ and $kA(P_1 - P_0)$).

We can show in a similar fashion, affine transformations preserve intersections. Assume the two lines intersect at u1 and u2 corresponding to the u parameters for lines L1 and L2, respectively. At intersection we have

$$\begin{aligned} &P_{L1(u1)} = P_{L2(u2)} \\ &P_0 + u_1(P_1 - P_0) = P_2 + u_2((P_3 - P_2)) \end{aligned} \tag{12}$$

Equation (12) can be rewritten while preserving equality as shown below.

Equation 13 signifies two important closure properties. It shows that both intersections, as well as relative

[5] We apply matrix algebra in above; for matrices A, B, C and scalars s and r we have sArB = srAB, and A(B + C) = AB + AC

locations of where intersections occur, are preserved under affine transformations.

$$\begin{aligned}&P_{L1(u1)} = P_{L2(u)}\\&A(P_0 + u_1(P_1 - P_0)) + B = A(P_2 + u_2((P_3 - P_2))) + B\\&(AP_0 + B) + u_1((AP_1 + B) - (AP_0 + B))\\&= (AP_2 + B) + u_1((AP_3 + B) - (AP_2 + B))\\&\tau(P_{L1(u1)}) = \tau(P_{L2(u2)})\end{aligned} \qquad (13)$$

4 Special Cases, Alternative Representation of Affine and Linear Transformations

While intersections are preserved under affine transformations, and hence linear transformations, angles are not preserved. For a special type of transformations, however, one can show that angles are preserved under the transformation. When composing transformations (performing successive transformations on objects) if the transformation is linear, the composition can be represented as matrix product. This is not the case when affine transformations are used.

In this part of the course, students are introduced to these discussion topics. We start with preserving angles under transformations. Since translation does not affect vectors, we restrict the discussion to vectors (with respect to vectors, translation acts as the identity matrix).

The concept of inner product is added to vector spaces to give a measure of distance and angles. Inner products are mappings from $R^n \times R^n$ to R. For orthogonal matrices[6], linear transformations preserve angle and length, the inner product is preserved.

First we discuss relation of inner product to matrix product. For two n-dimensional vectors represented as column matrices as $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})^T$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})^T$, the inner product is defined as

$$X_1.X_2 = \sum_{i=1}^{i=n} x_{1i} \times x_{2i} = X_1^T.X_2$$

$$= \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \end{pmatrix} . \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \end{pmatrix} \qquad (14)$$

Equation (14) relates inner product to matrix product. The dot on the left-hand side is represents the inner product operator while the dot on the right-hand side represents matrix product.

The inner product relates vectors length and angle between vectors as well where

$$X_1.X_2 = \sum_{i=1}^{i=n} x_{1i} \times x_{2i} = \|X_1\|\|X_2\|\cos\theta \qquad (15)$$

with $\|X_i\|$ representing the length of the vector and θ representing the angle between the two vectors. Note that

$$X_1.X_1 = \|X_1\|\|X_1\|\cos 0 = \|X_1\|^2 \qquad (16)$$

The above is used to verify that vector length is preserved under linear transformations with orthogonal matrix as follows.

To verify length is preserved, consider a vector X. We have

$$\begin{aligned}L(X).L(X) &= (AX)^T.(AX)\\&= X^T A^T AX\\&= X^T I_n X\\&= X.X = \|X\|^2\\&= \|L(X)\|^2\end{aligned} \qquad (17)$$

In above, A^T represents the transpose of A[7]. In addition, for orthogonal matrices the transpose of the matrix is its inverse.

Angles are shown to be preserved similarly. Let X and Y be two vectors. Consider L(X).L(Y) where the "." is the inner product.

$$\begin{aligned}L(X).L(Y) &= (AX)^T.(AY)\\&= X^T A^T AY\\&= X^T I_n Y\\&= X.Y\end{aligned} \qquad (18)$$

Equation (18) states that inner product is preserved for linear transformations where A is an orthogonal matrix. Since length is preserved and since the inner product is the product of the vector lengths and the cosine of the angles between the two vectors, the angles are preserved as well.

Composing transformations in computer graphics is to apply a sequence of transformations on a given object. When linear transformations are used, the sequence of geometrical transformations maps nicely to matrix product. Equation (19) is an example equation where an object X is transformed by linear transformations L1 followed by L2, … , Ln.

$$L_n(\ldots L_2(L_1(X)) = (A_n \ldots A_2 A_1)X \qquad (19)$$

[6] Matrix, A, is orthogonal if and only if the columns of A form an orthonormal basis for R^n.

[7] The transpose of product satisfies $(AB)^T = B^T A^T$

The result is more complicated when affine transformations are applied. For a sequence of n affine transformations we have

$$\begin{aligned}\tau_n(\ldots\tau_2(\tau_1(X))) &= (A_n\ldots A_2A_1)X + (A_n\ldots A_2B_1) + \cdots \\ &+ (A_nA_{n-1}B_{n-2}) + (A_nB_{n-1}) + B_n\end{aligned} \tag{20}$$

To normalize the transformations into a single format of a matrix product, homogeneous coordinates are used where n-dimensional space is extended to an (n + 1)-dimension with the (n + 1) component set to zero for a vector and to 1 for a point. This notation preserves the allowable closure operations in the affine space. In affine space: 1) subtraction of two points leads to a vector, 2) the sum of a point to vector results in a point, and 3) affine sum of points results in a point. These properties are preserved. With the above, an affine transformation of the form τ(X) = AX + B can be represented as a linear transformation as given in equation (21). Linear transformations are represented similarly with the n + 1 column replaced with the zero column.

$$\begin{aligned}\tau(X) &= \begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_1 \\ 0 & \cdots & 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_1 \\ \vdots \\ x_n \\ 1 \end{pmatrix} \quad or \\ \tau(X) &= \begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_1 \\ 0 & \cdots & 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_1 \\ \vdots \\ x_n \\ 0 \end{pmatrix}\end{aligned} \tag{21}$$

The equation shows the same representation for vectors and points with the exception of the (n+1) component/coordinate of X (1 for a point and 0 for a vector).

6 Conclusion

Transformations play very important role in computer graphics. Of the transformations used, two of the most common transformations are linear and affine transformations. In this paper we considered a unifying model and methodology of presenting the two transformations. Linear transformation work on vector spaces where both the domain and the range are vector spaces. Affine transformations work on extended domain and range where the space includes n-dimensional points in addition to vector spaces. The different properties preserved under both transformations were explored. In addition, the need to form alternative representation of affine transformation similar to linear transformation was presented.

References

[1] E. Angel. "Interactive Computer Graphics, A Top-Down Approach Using OpenGL, 6th edition". Addison Wesley, 2012.

[2] E. Angel. "OpenGL, A Primer, 2nd edition". Addison Wesley, 2005.

[3] P. Burger, D. Gillies. "Interactive Computer Graphics: Functional, Procedural and Device Level Methods". Addison Wesley, 1989.

[4] H. Farhat. "Transformations and Basis in the Computer Graphics Viewing Pipeline". Int'l Conf. Frontiers in Education: CS and CE, FECS'09, pp. 267-271, July 2009.

[5] J. Foley, A. Van Dam, S. Feiner, J. Hughes. "Computer Graphics Principles and Practice, 2nd edition". Addison Wesley, 1990.

[6] D. Hearn, M. Baker, W. Carithers. "Computer Graphics with OpenGL, 4th edition". Prentice Hall, 2011.

[7] F. Hill, S. Kelly. "Computer Graphics using OpenGL, 3rd edition". Prentice Hall, 2007.

[8] J. Hughes, A. Van Dam, M. McGuire, D. Sklar, J. Foley, S. Feiner, K. Akeley. "Computer Graphics Principles and Practice, 3rd Edition". Addison Wesley, 2014.

[9] D. Rogers, J. Adams. "Mathematical Elements for Computer Graphics, 2nd edition". McGraw Hill, 1990.

[10] D. Rogers. "Procedural Elements for Computer Graphics". McGraw Hill, 1998.

[11] A. Watt, F. Policarpo. "The Computer Image". Addison Wesley, 1998.

[12] A. Watt, "3D Computer Graphics, 3rd Edition". Addison Wesley, 2000.

Visualization and Computer Aided Design Techniques for Teaching Computer Hardware Design Course

J. Pang[1]
[1]Department of Electrical and Electronic Engineering and Program of Computer Engineering,
California State University, Sacramento, Sacramento, CA, U.S.A.

Abstract - *It is essential for undergraduate computer engineering students to have thorough understanding of the computer internal architecture. However, the study of computer hardware usually involves with a lot of abstract concepts, complex hardware structural interconnections and also dynamic hardware behavior. In order to invigorate students in efficient and active learning, effective technology integration with classroom instruction is required. This paper proposes three different visualization techniques for active learning including circuit diagram visualization, waveform visualization, and signal flow visualization. In addition, this paper presents how to use computer aided design method such as using a free open source logic design tool Logisim to assign projects to students so that they are able to integrate course contents, enhance problem solving skills, and also engage in active learning.*

Keywords: Active learning, computer engineering education, visualization, computer aided design

1 Introduction

Understanding of the internal architecture of microcomputers is the basis for undergraduate computer engineering students to obtain in-depth knowledge of the design and application of computers. Traditionally, real digital hardware was used inside classroom for teaching microcomputers [1][2][3] and it was helpful for students to understand usage of microcomputers [4][5][6]. However, usually real digital hardware and associated software tools are expensive [7]. Moreover, it is time consuming for students to conduct experiments which require wiring components together. Especially, it is frustrating for students to trace problems when there are hardware related errors, such as broken wires, wrong connections, stray inductance and capacitance, power bus noise, and so on. This may also distract students from efficient understanding of the internal architecture of microcomputers. A separate and dedicated electrical engineering circuit class is necessary for students to focus on electronic circuit issues.

Active learning strategies using visualization and logic design tools have been used by author of this paper to teach computer hardware design course to undergraduate computer engineering students at CSUS. The author has observed that when students study course materials, it is hard for them to associate a lot of static texts and abstract concepts with complex internal connections of microcomputers. It is also hard for them to imagine how digital signals propagate inside microcomputers to operate different functional units. Representing information into graphic format through visualization techniques, and also using computer aided logic design tools for projects can concisely demonstrate microcomputer architecture related ideas. It is beneficial for students to enhance their involvement and comprehension of course materials, to reduce stress, to get motivated and also to have better insight into the internal architecture of microcomputers

This paper looks at the roles and implementation details of using visualization techniques and computer aided logic design tool for teaching computer hardware design course. Part 2 describes how to use multiple visualization techniques to represent information inside classroom. Part 3 presents examples of using Logisim tool [8] for student projects which help them study the internal architecture of the microcomputer. Part 4 draws conclusions of this work.

2 Use of visualization techniques in class

Visualization techniques help students understand microcomputer architecture details at various levels of complexity and they provide important supporting roles to instructors inside classroom. Three different visualization techniques are proposed in this paper.

2.1 Circuit diagram visualization

A circuit diagram representation of computer concepts enables students to approach course material in more concrete way, and to visualize abstract behavior of computer hardware architecture more clearly and effectively.

Take memory hardware which is used to store data as one example. In general, the *A1* and *A0* signals represent address of the memory, the *CS* signal enables the memory, the *RD* signal allows reading memory data, and the *OE* signal allows the memory to output data onto the data bus. Such descriptions can be visualized through the circuit diagram shown in Figure 1 which is a 4x2 memory circuit. The analysis of circuit diagram allows students to explore the functionality of the circuit, and to enhance their understanding of the memory behavior.

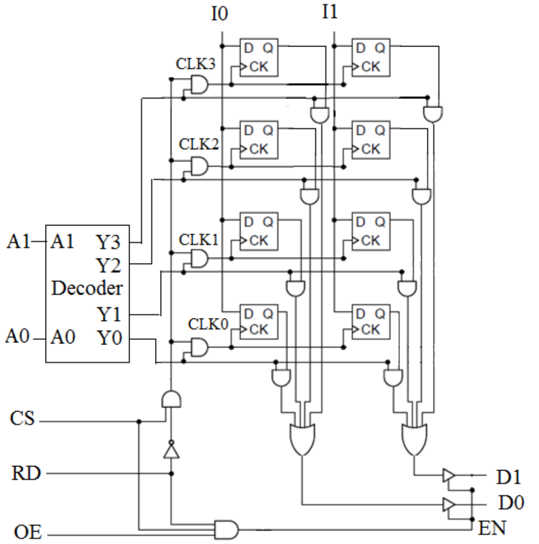


Figure 1. 4x2 memory circuit

For example, in Figure 1, when the *CS* signal is logic 0, it results in all logic 0s for the *CLK0*, *CLK1*, *CLK2*, *CLK3*, and *EN* signals, which are used to generate the clock pulses and also to enable the output tri-state buffers. Because these signals are disabled, new data inputs won't be written into D Flip-flops, and also the output data bus will be in high-impedance mode. Through such analysis, students will be able to understand why the *CS* signal can disable the memory.

Instructors can use circuit diagrams to stimulate in-class discussion, to engage students in active learning and to initiate a collaborative effort among students for finding answers and solutions to the functionality of circuits.

2.2 Waveform visualization

Waveforms can help students find relationship among multiple signals, and visualize signal patterns. Figure 2 shows the simulation waveform for the memory circuit in Figure 1.

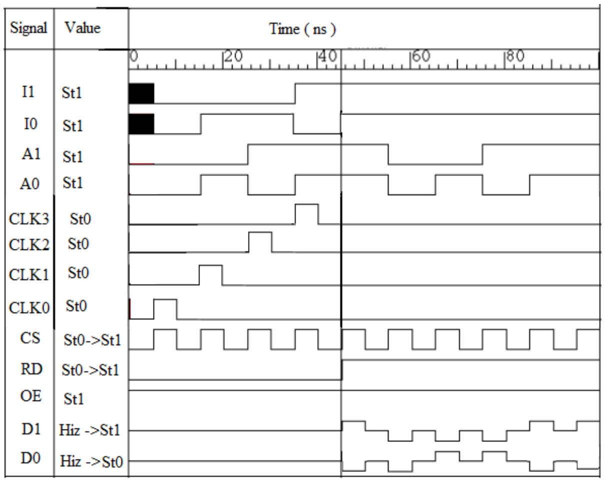


Figure 2. Memory simulation waveform

In Figure 2, it is obviously to see that the *CLK0*, *CLK1*, *CLK2*, and *CLK3* pulses are generated at different time. For example, the *CLK3* pulse is generated due to the input of *CS* pulse from 35 ns to 45 ns. At this time, the *A1A0* signals point to memory address 3, and the memory data input signals *I1* and *I0* are binary "10". Because *RD* is logic 0, it allows binary data "10" to be written into the memory address location 3. Here, multiple signals show joint actions in order to write data into memory. In another word, to write data into memory location 3, the *CS* signal needs to be logic 1 to enable the memory hardware. The address signals need to be set at value 3, and the *RD* signal needs to be logic 0 to allow writing to occur.

From 45 ns to 50 ns in Figure 2, because the *RD* and *CS* signal are activated with logic 1s, the memory cells at address 3 can output data "10" onto the data bus *D1D0*. This shows how memory reading works. It also proves the previous writing was successful.

In general, waveforms can be used to promote hardware functionality analysis and extend students' understanding of key concepts of computer hardware.

2.3 Signal flow visualization

To analyze the dynamic behavior of interconnected computer system and also to provide a more complete view of how computer hardware works, a signal flow diagram is beneficial for students to visualize the hardware complexity in a more comprehensible way. Signal values can be marked on the hardware diagram for different case studies as displayed in Figure 3. Figure 3 shows the Mic-1 microprocessor [9] consisting of data path and control parts interweaved together.

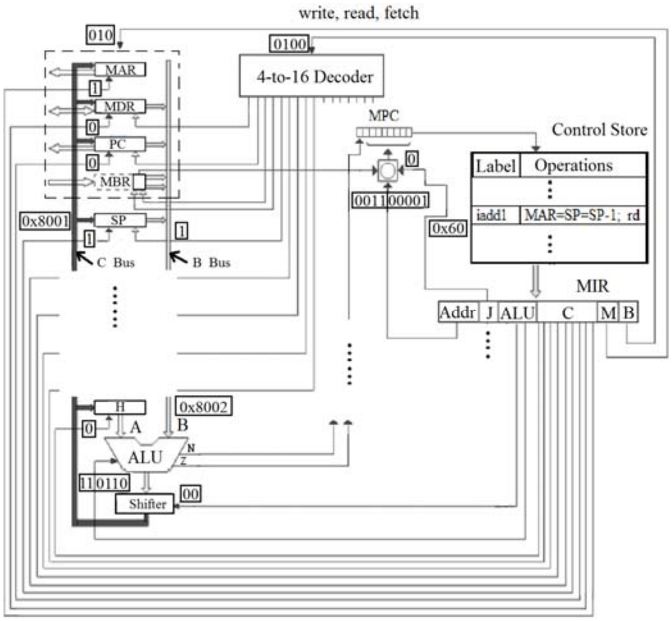


Figure 3. Signal flow on Mic-1 microprocessor

Mic-1 architecture implements Integer Java Virtual Machine (IJVM) instruction set. Figure 3 shows signal flow values marked inside black rectangles for iadd1 micro-operation. IADD instruction is one of IVJM instructions, which pops two words from the stack memory, and then push their sum into the stack. To implement IADD, three micro-operations of iadd1, iadd2, and iadd3 are needed. iadd1 is used to read next-to-top stack data. The opcode 0x60 for IADD points to the control store memory address where iadd1 micro-operation is stored. The iadd1 micro-operation is fetched into the micro-instruction register *MIR*, which controls hardware in the data path.

In Figure 3, the four bits of “B field” inside the *MIR* register are equal to “0100” which are connected with a 4-to-16 decoder. This enables the 5th output of the decoder so that the SP register can output content “0x8002” onto the B bus.

The “ALU field” of the *MIR* register has totally 8 bits. Two bits are “00” which disable the shifter circuit from shifting. And the other 6 bits are “110110” which choose ALU function of B - 1. As a result, the B bus data “0x8002” is subtracted by 1 through ALU and then generates 0x8001 on the C bus, which is the next-to-top stack address value.

The “C field” of the *MIR* register outputs binary numbers which are used as write enable signals for different registers connected with the C bus. Since only the write enable signals of *MAR* and *SP* registers are logic 1s, the *MAR* and *SP* registers are activated. As a result, both *MAR* and *SP* registers will be updated by the next-to-top stack address.

Since the “M field” of the *MIR* register outputs binary number “010” for write, read and fetch control of the main memory, the main memory will be set to read mode. In another word, the Mic-1 processor is going to read next-to-top stack data from the main memory and store it into the memory data register *MDR*.

After iadd1 micro-operation is processed by data path hardware, Mic-1 is going to process next micro-operation of iadd2. The logic 0 values of the “J filed” in the *MIR* register allow the *MPC* register to choose the next micro-instruction address from the “Addr filed” of the *MIR* register, and it is “001100001” (0x61). The *MPC* register stores the control store memory address for the next micro-operation. At the address 0x61 location of the control store, the micro-operation iadd2 machine code is stored. Then the whole processing steps will be repeated to implement the new micro-operation.

Signal flow helps explain cause and effect of different dynamic actions inside the computer hardware with the support of data. It can engage students in the active learning process, and also increase their attention and focus. Moreover, it helps students understand context in a tangible way.

Signal flow data can be fully provided by instructors, or partially by instructors and partially by students through questions and problem solving inside the classroom. Such practice increase students’ curiosity for course contents, and also promotes meaningful learning experiences among students.

3 Computer aided design projects

Project assignments using computer aided design tools help students integrate topics covered in the course, and engage them in active learning instead of just memorizing knowledge transferred from instructors in class. As a free and user friendly open source educational tool, Logisim [8] allows students to design and simulate digital logic circuits. This tool has been used by author of this paper to teach computer hardware design course to undergraduate computer engineering students at CSUS.

Figure 4 shows one project idea of using Logisim. It requires students to create the register logic sub-circuit to be connected with the data bus, the ALU, the control store, and the main memory of the Mic-1 processor.

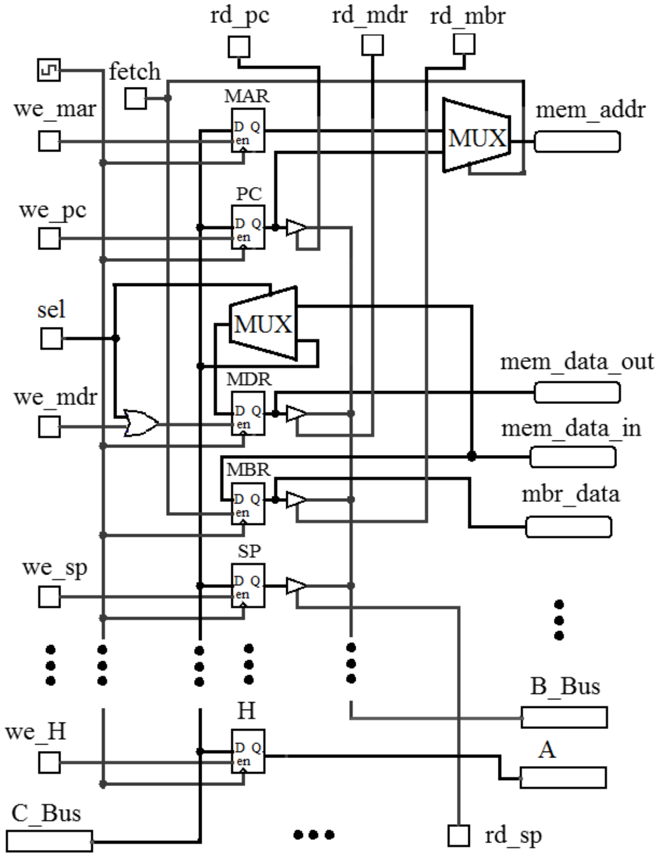


Figure 4. Registers and data bus

The ALU needs two operands A and B. The operand A comes from the register *H*. The operand B comes from one of those registers with read enable controls and also connected with the data bus *B_Bus*. Every register connected with the data bus *C_Bus* has write enable control for storing the ALU result. For example, the *rd_pc* is the read enable control for the *PC* register, and the *we_pc* is the write enable control for the *PC* register. In addition, registers *MAR*, *PC*, *MDR*, and *MBR* are used to interface with another sub-circuit of the memory block.

In Figure 4, when the *fetch* signal is logic 0, it selects the *MAR* register output to be the memory address *mem_addr*. When the *fetch* signal is logic 1, it selects the *PC* register output to be the memory address, and also it allows reading instruction *mem_data_in* from the main memory into the *MBR* register. The *MDR* register has write enable controls from *we_mdr* and *sel*. The signal *we_mdr* allows the *C_Bus* data to be written into the *MDR* register. The signal *sel* allows the memory data *mem_data_in* to be written into the *MDR* register. The signal *mem_data_out* comes from the *MDR* register output which can be written into the main memory. The output *mbr_data* can be used as one of inputs to the CONTROL_STORE subcircuit to determine the next microinstruction address.

Moreover, Logisim has built-in RAM which can be configured as the Mic-1 memory space to hold the main IJVM instructions, local variable, stack, and constant pool data. Logisim also has built-in ROM which can be configured as control store to hold micro-instructions. Students are required to design appropriate micro-instructions to be loaded into the control store to interact with the data path. In conclusion, the whole Mic-1 processor can be created, and simulated by using the Logisim tool. Modification of the Mic-1 is also possible which may stimulate students for more creative design ideas.

4 Conclusions

Three different visualization techniques are discussed in this paper including circuit diagram visualization, waveform visualization, and signal flow visualization. These techniques allow students to study computer hardware in different perspectives, to visualize computer hardware concepts in more tangible ways, and to improve their learning experience. In addition, computer aided design projects stimulate students to think creatively, to integrate course contents, and to enhance their problem solving skills. Combining different visualization techniques for instruction inside classroom, and also using computer aided design project assignments for students outside classroom, instructors are able to teach computer hardware course more effectively and efficiently, and also engage computer engineering students in active learning.

5 References

[1] S. Vijay Iyengar, Larry L. Kinney. “A Design-Oriented Microprocessor Laboratory”; IEEE Transactions on Education, Vol. 24, Issue 1, Pages 43 – 46, Feb. 1981.

[2] James E. Cross, Raphael A. Soetan. “Teaching Microprocessor Design Using the 8086 Microprocessor”; IEEE Conference Proceedings on Southeastcon, Pages 175 – 180, Apr. 1988.

[3] A. J. D. Casaca, J. V. Woods. “Teaching Digital-Systems Design with a Microprocessor-Controlled Patchboard”; IEE Proceedings on Computers and Digital Techniques, Vol. 130, Issue 3, Pages 75 – 82, May 1983.

[4] Maosheng Wu, Ximeng Wu, “Design of the Single-chip Microcomputer Experiment Teaching System for Application”; 2011 International Conference on Electrical and Control Engineering, Pages 649 – 652, Sept. 2011.

[5] S. Arunaganesan, J. Adhavan, G. Shivakanth Reddy, M. Venkatesan. “Data Acquisition System Based on 8051 Microcontroller for Cutting Tool Temperature Measurement”; 2013 IEEE International Conference on Computational Intelligence and Computing Research, Pages 1 – 4, Dec. 2013.

[6] Hussam M. Elbehiery, M. Ghada Abdelmouez. “Implementation of New Symmetric Ciphering on ATMEGA 32”; 2010 International Computer Engineering Conference, Pages 1 – 8, Dec. 2010.

[7] Tom Stoltz, Mark Paulik, Nizar Al-Holou. “A Microcontroller Laboratory Hardware Platform for the Academic Environment: The UDM-EVB”; Proceedings of 35th ASEE/IEEE Annual Conference on Frontiers in Education, Session S2G-1 – S2G-6, Oct. 2005.

[8] Carl Burch. “Logisim: A Graphical System for Logic Circuit Design and Simulation”; Journal on Educational Resources in Computing, Vol. 2, Issue 1, Pages 5 – 16, March 2002.

[9] Andrew S. Tanenbaum, Todd Austin. “Structured Computer Organization”. Pearson publisher, 2012.

It's a NUI World: Incorporating Natural User Interfaces in an HCI course

A. Dheivya Thiagarajan[1], **B. Caitlin Hurley**[1], **C. Amanreet Bajwa**[1], **and D. Cyndi Rader**[1]
[1]Electrical Engineering & Computer Science, Colorado School of Mines, Golden, CO, USA

Abstract - *Tablets and mobile phones have become ubiquitous, and computer science education needs to reflect this trend. In particular, with the escalating demand for programmers with the skills to create mobile and tablet apps, it is important for our students to learn how to develop effective user interfaces that take advantage of the unique capabilities of these new technologies. How to accomplish that goal is an open question. Some practitioners have suggested that the new devices support a different type of interaction called a Natural User Interface (NUI). In this paper, we report our experience incorporating NUI guidelines into a human-computer interaction course. Despite some challenges with the Android development environment, students generally saw the value and were able to apply at least some rudimentary NUI principles to their app designs.*

Keywords: Human Computer Interaction (HCI), mobile programming, Android

1 Introduction

The popularity of mobile devices is currently skyrocketing. As a result, it is increasingly important to incorporate mobile design principles into user interface courses. In this paper, we present one approach based on what has been dubbed the “natural user interface” or NUI. In general, the goal of a NUI is to “enable users to interact with computers in the way we interact with the world.” [6]

NUIs encompass a number of modalities such as speech, gesture, touch and 3D manipulation, and therefore have potential applicability beyond just phones and tablets. But gestural interfaces, just as mouse interactions, must be learned. Some HCI researchers have therefore questioned whether NUIs really provide a superior interaction paradigm. One critique is that users don't really interact naturally, which may vary by user, but must use a predefined set of gestures designed in a lab [10]. Others claim that gestures are not that easy to learn or remember, and are not understood consistently across different cultures [12]. Furthermore, it is critical that existing knowledge of what makes an interface effective, such as perceived affordances, feedback, consistency, scalability, discoverability, reliability, and non-destructive operations, not be ignored in the rush to create the next exciting technology [13].

On the other hand, several studies have shown that these user interfaces actually *are* natural because they are intuitive. For example, researchers who used a touch interface to create electronic medical records found that less training was required by the health-care workers using their system [4], and voice recognition as the primary form of input for a search utility has been shown to be more natural on a small mobile device than trying to type out a query [5].

Our goal in this paper is twofold: a) to gauge students’ attitudes toward NUI as a guiding principle and b) to present and evaluate one approach to inserting NUI concepts into an existing user interface design course.

2 Related Work

A number of universities have incorporated mobile application development into their curriculum. Capitalizing on the popularity of mobile and tablet devices, a number of researchers have successfully used Android development to spark enthusiasm for computer science in introductory courses [e.g., 8]. Other researchers have used the accessibility of the Android platform to teach operating system concepts [2], embedded systems [11], software engineering [14], advanced programming in Java [15], and data structures [1]. Incorporating mobile development into a course, no matter the underlying motivation, has generally proven to have positive results.

Given the prevalence of mobile development in industry, it is imperative that HCI courses also be adapted to address the unique constraints and opportunities provided by cell phones and tablets. One researcher who included Android development in an HCI course reported a positive impact on the students [9]. In that course, students completed two major projects, one web-based and one mobile-based. To accommodate users with a wide range of abilities, the apps created by the students incorporated speech input and feedback. Differences between the web and mobile applications were used to illustrate various aspects of mobile application design, such as the need to minimize typing and accommodate various screen sizes on the mobile devices.

3 Approach

Interfaces that follow the NUI paradigm are becoming prevalent, and therefore students in an HCI course should receive instruction in this style of interaction. But it is important to balance NUI ideas with traditional HCI topics, and to help students realize that just incorporating gestures into an app does not, necessarily, make that app easy to use.

For our UI design class, we used the Android platform as a teaching tool for NUIs and gestures, as these features are built-in and are a dominant feature of the phone. To address concerns that a course emphasizing NUI might neglect existing HCI design principles, Android development was used as only one component of the course. Students also learned a goal-directed design process [3] and explored the relationship between cognitive processes and user interface design [7]. NUI principles were used to guide the mobile development.

Developing for Android, and mobile devices in general, differs from developing for a desktop or web application in several ways. Two obvious differences are the amount of space available for the display and the use of a touch interface which incorporates tap, pinch, and flick gestures, voice commands, and even movement of the phone to communicate with an application (e.g., shaking the phone to shuffle items in a game such as *Words With Friends*). To provide students with a vocabulary and set of guiding principles for designing mobile applications, we decided to focus on natural user interfaces as defined in the book *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, by Daniel Wigdor and Dennis Wixon [16]. The next two sections describe the NUI topics we covered and the related activities.

3.1 Topics

The selected textbook [16] includes a grand vision for NUI and numerous suggestions for creating programs that might realize that vision. For our course, specific topics that were stressed include:

3.1.1 Fat finger problem

Two issues result in a "fat finger" problem: a) unlike a mouse click, which occurs at a specific pixel, fingers cover a larger area, and b) the item to be pressed is hidden by the touching finger. To address this problem, the authors recommend making the targets larger and using "iceberg targets" which respond to a larger area than just the visual icon/button.

3.1.2 Immediate feedback

With touch devices, users tend to attribute a lack of feedback to the application breaking, rather than a wrong gesture. The authors suggest that the application should provide immediate feedback to every input, even incorrect ones, so that users know the device is working and what result each gesture had. This feedback may be visual or haptic (e.g., vibration).

3.1.3 Anatomy of a gesture

Touch gestures consist of registration (e.g., place two fingers on an object), continuation (e.g., move fingers to stretch object), and termination (e.g., lift fingers from the device). The authors discuss a number of guidelines to remove the ambiguity inherent in many touch screen interfaces.

3.1.4 New primitives

A "primitive" is a building block for an interaction language. For example, point and click are two primitives for a mouse-based system. The set of primitives for touch-based systems are currently small (e.g., pinch to zoom in and out, slide to scroll, etc.). New primitives for touch interaction need to be designed to facilitate NUI. For example, instead of clicking to check a box, a more natural action on a touch Screen may be some type of crossing primitive.

3.1.5 Super realism

One advantage of touch is that it enables direct interaction with objects on the screen. Super realism enhances the experience by allowing users to interact in ways that would not be possible in a physical environment, for example by simultaneously allowing different perspectives, stretching objects that are normally fixed, etc. Single finger drag-and-flick movements of on-screen objects is one example of incorporating super realism into an application.

3.2 Activities

NUI topics were explored via reading assignments, in-class exercises, and programming homework. The four reading assignments covered 15 of the 30 chapters in the book. The purpose of these assignments was to ensure that students read the chapters and were prepared to do the in-class exercises.

For the in-class exercises, students typically worked in pairs to apply principles from the text to the design of an app, such as applying the guidelines for a "Social NUI" to a crime scene simulation or fashion design app. The in-class exercises were done on paper. Students were encouraged to be very creative in the design exercises, and some of the solutions were shared with the class.

To give the students an opportunity to implement what they were learning, a separate set of exercises were developed for the Android platform. Most students in the class had no prior Android programming experience, although many were concurrently enrolled in an *Introduction to Android* elective course. An additional impediment, which was not known at the beginning of the semester, was that the Android emulator would not work in the computer lab where the class was held. As a result, most of the students needed to learn basic Android fundamentals with minimal classroom support.

There were five small exercises related to Android programming that were done either individually or with a partner, and one larger team project. The exercises served two purposes: 1) to help students learn Android programming and 2) to apply one or two NUI principles. Each exercise included links to a tutorial that the students would complete along with a small extension. To avoid duplicating material from the *Introduction to Android* course, these exercises focused primarily on the user interface and avoided content such as threads, databases, and other topics needed to create full-fledged Android applications.

Table 1. Android topics and NUI exercises

Android/ NUI Topics	Exercise
Basic forms / None	Accept person's name, display hello message when button is pressed.
Gestures/ New primitives	Record two gestures, respond to one with a toast/message, launch a new activity for the second.
Single taps and flicks/ Super realism	Display six "target" icons and a projectile. For a single tap, move the projectile to that location. For a flick, send the projectile toward the nearest target in that direction.
OnTouchEvent/ Fat fingers	Display six targets. Highlight the one that is clicked. Include an "iceberg" parameter that increases the clickable area around each icon.
Drag-and-drop/ Anatomy of a gesture	Display three icons (oranges) on the screen. When the user selects one (registration), the image changes. The user drags the orange to the basket (continuation). When the orange is "dropped" in the basket (termination), the image again changes.

Table 1 provides a list of the exercises, including a very brief description and the Android and NUI topics the exercise was intended to reinforce or teach.

For the team exercise, students created a Math Tutor app to allow young students to practice simple addition and subtraction. The students were given one and a half weeks to complete their apps. The project specification required students to include at least four addition and/or subtraction problems, to provide feedback on the number of problems solved, and to use flicks, gestures, or some other type of engaging interaction (i.e., not just a simple form).

Although mobile development is currently a hot topic, NUIs are in their infancy, and we should ultimately strive toward making technology as "natural as using the hand to scratch an itch." [10]. So in addition to the design exercises and Android homework, students researched and did 20-minute presentations on topics related to UI and/or NUI. Many of these focused on new technologies such as eye tracking, near field communication, tactile and 3D interfaces, and Leap motion devices.

4 Evaluation

In this paper we are concerned with students' perception of the relevance of NUI and their acquisition of skills that would enable them to create effective NUI interfaces themselves. To assess students' attitudes, the students completed a survey toward the end of the semester. The survey included eight Likert-scale questions, with two that address the relevance of NUI, five to gauge the effectiveness of our teaching approach, and two related to students' own interest or competence. Two open response questions were used to gather additional feedback related to the textbook (not reported here) and the course. The survey was completed by 30 students (91% response rate).

To gain more insight into students' acquisition of NUI skills, we also did an analysis of the Math Tutor exercise.

4.1 Importance of NUI principles

The survey revealed that a majority of the students recognized the importance of learning NUI principles. Approximately 77% agreed or strongly agreed that software that adhered to NUI principles will become prevalent in the near future (Figure 1).

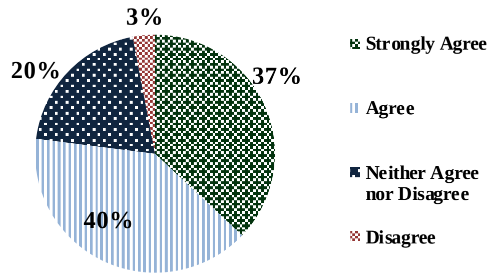


Figure 1. Software that adheres to NUI principles will become prevalent in the near future

When asked whether it was important for students in a UI Design course to be introduced to NUI principles, 90% of the students either agreed or strongly agreed (Figure 2). It should be noted that, since the only mobile framework used in this course was NUI, this question might more accurately reflect students' opinion that addressing mobile user interfaces is an important topic.

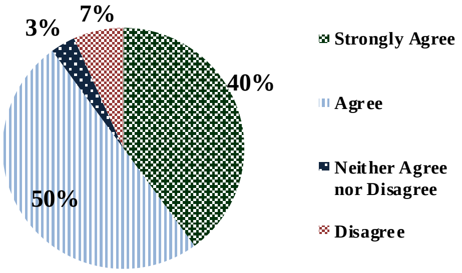


Figure 2. It is important for university UI design courses to introduce students to NUI principles.

4.2 Evaluating the course

Given the importance students ascribed to learning NUI, it was important to determine whether the structure of our course was effective for that purpose. We gathered feedback about both the textbook and the exercises. Questions and results are listed below.

4.2.1 The textbook explains NUI concepts effectively

Since the structure of the course relied on students reading the text to prepare for in-class exercises, it was critical that the book provided sufficient background and was easy to read and understand. The majority of students in the course were satisfied with the text, as 83% agreed with this question (23% strongly agreed, 60% agreed).

4.2.2 The NUI exercises helped to reinforce the concepts from the NUI textbook

Each chapter of the textbook included guidelines that *must*, *should* and *could* be followed to achieve the effect described in that chapter. These guidelines provided both concrete suggestions and high-level goals. For example, a *must* principle for super realism is to "Pre-buffer content, provide a transition, or use other mechanisms to make sure that every touch receives an immediate and meaningful response." A *should* principle states that "In creating this feeling of life, make sure that the behavior is subtle to avoid being annoying or distracting. Do not cause the applications' state to change; instead, change only background and graphical elements." [16, pp. 49-50]. We covered 15 chapters from the book, and each chapter ended with as many as 17 principles. Clearly this was too much material to cover comprehensively in a one-semester course, but our goal was to provide an opportunity, via the exercises, for students to put at least a few of the ideas into practice.

The majority of students either agreed (50%) or strongly agreed (10%) that the NUI exercises were effective. Five students (17%) disagreed, and seven students neither agreed nor disagreed. No student strongly disagreed.

4.2.3 The tutorials for each NUI exercise provided sufficient background to enable me to complete the NUI exercises

As mentioned, most students taking this course were new to Android programming, and for many students this was their only exposure. Each exercise therefore began with an online tutorial that students could modify in order to apply a NUI guideline. Informally, it was clear from the number of student emails that this was quite challenging for a few of the students. Although a majority of students did feel that the tutorials provided sufficient background (3% strongly agreed and 57% agreed), quite a few disagreed (17%) or strongly disagreed (7%).

It is quite likely that the students who agreed were concurrently taking the *Introduction to Android* course. Providing better training and support for students with no other exposure to Android will be critical for future iterations of the course. At the very least, having a working emulator available in the lab would be essential. In fact, an additional question asked whether the Android programming should *only* be included in the UI Design course if the Android emulator works in the classroom. The response to this question was mixed, as 50% of the students disagreed or strongly disagreed, 20% were neutral, and 30% agreed or strongly agreed.

Spacing the exercises throughout the semester, rather than in just one segment, could also provide more time for students learning Android to feel more confident and be able to focus on the NUI concepts, rather than struggling to just get programs to execute.

4.2.4 Compared to this year, how much emphasis should the UI Design course place on Android programming

An additional question asked students whether Android

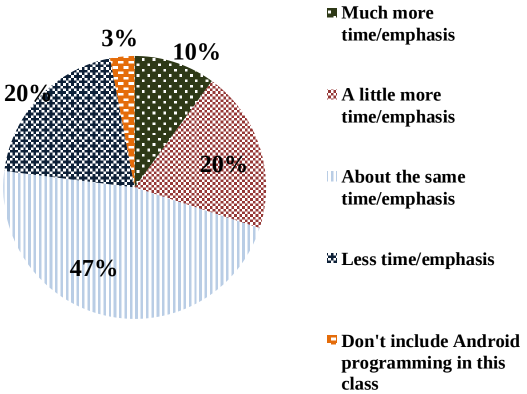


Figure 3. Compared to this year, how much emphasis should the UI Design course place on Android programming?

programming should be emphasized more, less, or the same amount in future iterations of the course. Close to a majority (47%) thought the amount was about right. It is quite possible that those who wanted less or no time struggled more due to the lack of classroom support for learning the basic Android programming. As shown in Figure 3, the response to this question was also mixed, as 30% preferred more time, but 20% desired less and 3% wanted no Android programming at all.

4.3 Personal Attributes

Two additional survey questions probed students' personal goals and feeling of competence.

4.3.1 I would personally like to develop applications that adhere to NUI principles

When asked whether they would personally like to develop applications that adhere to NUI principles, 83% either agreed or strongly agreed, 7% disagreed, and 10% had no strong opinion or were not sure (Figure 4).

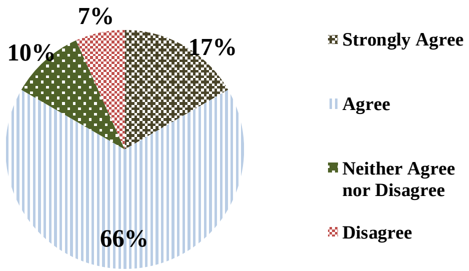


Figure 4. Would you personally like to develop applications that adhere to NUI principles?

Another indication of students' interest in Android programming came from the final projects. This was an open-ended assignment in which students worked in teams of two to five students to create projects of interest. The primary learning objective for this project was to practice goal-directed design [3]. Although Android programming was not required, all of the teams (12 in total) elected to do an Android app for their final project. Two of the teams chose to extend their Math Tutor games. Other projects included a mobile shuffle board, a logic gate simulator, a seating arrangement app, a to-do list app that could synch with the Google calendar, an algebra physics solver, a hike recorder, a Yahtzee game, and two recipe/shopping apps. Although not analyzed in detail, based on the in-class presentations it was obvious that a number of the projects made good use of NUI principles. For example, in the Yahtzee game, users could shake the phone to roll the dice.

4.3.2 I am confident that I can develop applications that adhere to NUI principles.

When asked if they thought that they could develop applications adhering to NUI principles, 70% either agreed or strongly agreed, 10% either disagreed or strongly disagreed, and 20% neither agreed nor disagreed. This result seems promising, given the issues many students had with learning Android programming.

4.4 Open-ended feedback

The students were given the opportunity to provide other suggestions/feedback related to the use of Android programming and/or NUI principles in the course. In general, students thought that Android was effective for teaching NUI concepts, although some students suggested that ultimately other innovative technologies would become more prevalent. A few students indicated that the assignments were too difficult, especially with the lack of support in the lab. Quite a few students mentioned the timing of the Android assignments. Between the readings and the programming, the students had a heavier than normal workload during the Android segment of the class, which was quite stressful for some students.

5 Math Tutor

For the course itself, students' Math Tutor apps were graded on the specific requirements listed earlier in this paper (i.e., include at least four problems, keep a total of number of correct answers, use some type of NUI interface). For this paper we also judged their apps based on *how well* they were able to follow the NUI guidelines. This analysis is, by definition, somewhat subjective.

5.1 Fat fingers and Iceberg targets

About half of the teams chose to use icons that the kindergartners would drag and drop. Every group that chose this option successfully handled the fat finger problem. As long as the user's finger is in the general area of the object, it will be selected.

5.2 Gesture detection

The other half of the teams chose to have their application recognize the number after the student "writes" it on the screen. These teams thought that it was important for their students to not only recognize the numbers but also practice writing them. These teams had to handle gesture detection: how does the app know what number the user is writing? The Android gesture detector works by comparing user actions to recorded gestures, so the students had to associate gestures with specific numbers. This creates problems when a number can be written in different ways, such as starting at the top of a 1 and drawing down or starting at the bottom of the 1 and drawing up. Gesture recognition is even more challenging when a number requires two gestures, such as entering a negative number. The system must determine if the two gestures were related or if there was a second guess. By recording multiple gestures for each number, most of the teams were able to handle the case where the number could be written in different ways. On the other hand, only one group was able to handle double digits or gestures that contained two strokes (e.g., some representations of the number 4), and one team only recorded each digit once, which made their app more difficult to use.

5.3 Immediate feedback to every input

For touch interfaces it is especially important for the user to receive feedback for every gesture, even if it is not correct. All teams provided feedback when students answered a problem correctly. Teams that chose the “writing” option generally did a good job of providing feedback by using a "toast" to display what number the app recognized (if any), along with the indication of correct/not correct. Most of the other teams, however, provided feedback for correct actions but nothing else.

For example, one team made the very engaging app shown in Figure 5. This application shows the math problem and provides three bananas, each associated with an answer. The goal is to drag the correct answer to the monkey so he can eat it. When a correct answer is dragged to the monkey, a green checkmark will appear; for an incorrect answer a red 'X' will appear. If the user does not drag the banana to the monkey, however, the app does not give any feedback on what the correct action should be. For example, if the user taps the banana or drags the banana to the ‘?’ mark following the math problem, there is no indication that the action was invalid.

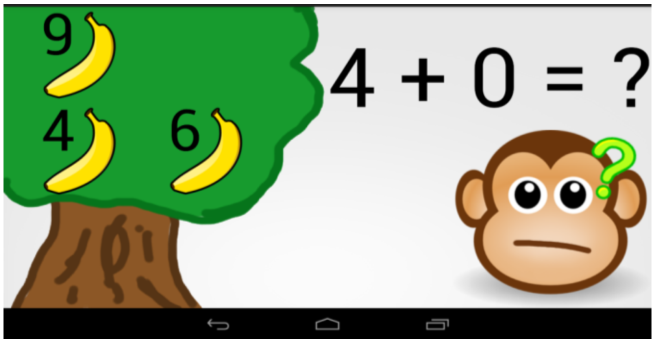
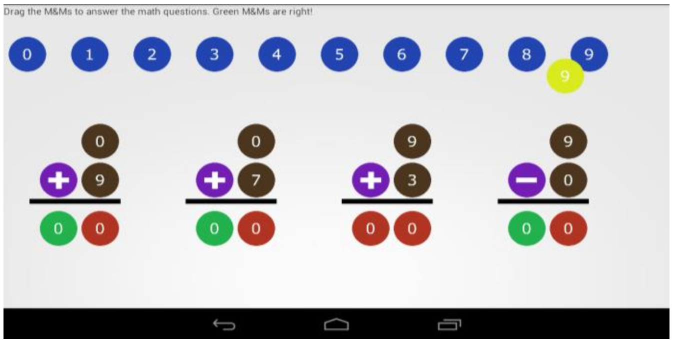


Figure 5. Drag the banana representing the correct answer to the monkey.

5.4 Anatomy of a gesture

The three common stages that make up the anatomy of a gesture are registration, continuation, and termination. This concept does not apply to teams that used the "writing" method, but two of the other teams did apply this guideline. For example, the app shown in Figure 6 allows users to drag M&Ms to answer math questions. Brown M&Ms represent the math problems, green indicate correct digits, and red are incorrect. The goal is for users to drag a blue M&M to the correct spot. As soon as the gesture is initiated (the user touches an M&M) it turns yellow. The M&M’s stays yellow until the gesture is finalized (the user releases the M&M), where it either turns red or green.



Drag the M&Ms to answer the math questions. Green M&Ms are right!

Figure 6. Drag the correct blue M&M to the solution spot

5.5 Super realism

Many of the apps that used drag and drop incorporated super realism. In the app shown in Figure 7, users must answer math problems by moving the correct number of boxes from the side of the screen to the "Solve" box in the middle. The figure shows a correct answer, in which the user has moved 10 boxes to represent the sum 5 + 5. This app mimics some real-life situations such as a grocery store that would restock its shelves by moving items from the back room. It incorporates super realism by allowing users to "flick" the boxes instead of having to drag them the entire way.

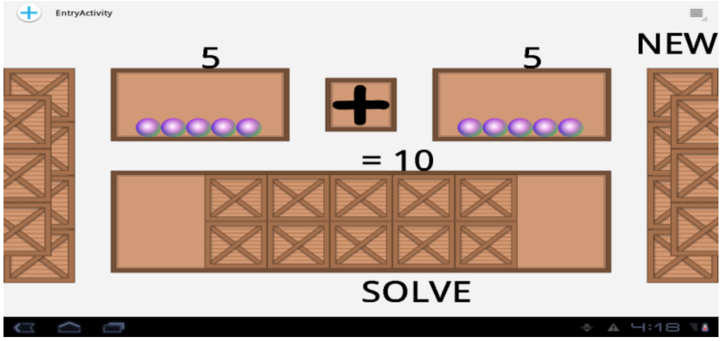


Figure 7. Move correct number of boxes from the side of the screen to the middle.

5.6 Overall Analysis

Despite the lack of time and training, most students were

Table 2. NUI principles used in Math Tutor apps

Principle	Number of teams
Fat fingers and iceberg targets	5
Gesture detection	6
Immediate feedback	11
Anatomy of a gesture	2
Super realism	6

able to effectively apply some of the NUI concepts they learned to their applications. All of the groups were able to create a tool that would be helpful in assisting young students practice arithmetic. Table 2 provides a count of the number of teams that applied each NUI principle. Note that this table includes teams who attempted a principle, even if there was room for improvement.

6 Conclusions

Based on student feedback and the level of interaction in the projects they created, we believe this was a successful first attempt. But there is, of course, clear room for improvement. The first imperative is to provide better support for students to learn basic Android programming. It is a testament to the inherent appeal of mobile devices that students were able to successfully create engaging apps, despite the lack of classroom training. In the open-ended comments, a number of students mentioned the short amount of time allocated for completing the Android assignments. These concerns could easily be addressed by interlacing Android assignments with other course material.

The survey results indicated that 10% of the students were not confident they could create applications that adhere to NUI, and 17% did not think the NUI exercises helped to reinforce the NUI concepts. To address these responses, it would be good to refine some of the exercises and/or add new exercises to relate to additional topics from the book. For example, the programming exercises were fairly prescriptive. A more productive approach might be to identify a number of NUI guidelines and a context, then ask the students to code their own creative solutions to meet those guidelines. This approach was used for the in-class design exercises, but not the programming assignments.

7 References

[1] Allevato, A. and Edwards, 2012. S. RoboLIFT: Engaging CS2 Students with Testable, Automatically Evaluated Android Applications, In Proc. of the 43rd ACM Technical Symposium on Computer Science Education, (Raleigh, NC, February 29–March 3, 2012), ACM, 547-552.

[2] Andrus, J. and Nieh, J, 2012. Teaching Operating Systems Using Android. In Proc. of 43rd SIGCSE Technical Symposium on Computer Science Education, (Raleigh, NC, February 29–March 3, 2012), ACM, 613-618.

[3] Cooper, A., Reimann, R., and Cronin, D., 2007. About Face 3: The Essentials of Interaction Design, Wiley, Indianapolis, IN.

[4] Douglas, G. P., Landis-Lewis, Z., and Hochheiser, H., 2011. Simplicity and usability: Lessons from a Touchscreen Electronic Medical Record System in Malawi. Interactions Vol. 18 No. 6, (November + December 2011), ACM, 50-53.

[5] Hearst, M. A., 2011. 'Natural' Search User Interfaces, Communications of the ACM, vol. 54, no. 11 (November 2011), ACM, 60-67.

[6] Jain, J., and Lund, A., 2011. The future of Natural User Interfaces; CHI 2011, (Vancouver, BC, Canada, May 7–12, 2011), ACM, 211-214.

[7] Johnson, J., 2010. Designing with the Mind in Mind. Morgan Kaufmann, Burlington, MA.

[8] Kurkovsky, S., 2009. Engaging Students through Mobile Game Development, In Proc. of 39th SIGCSE Technical Symposium on Computer Science Education (Portland, OR, March 12-15, 2008), ACM, 44-48.

[9] Loveland, Susan. Human Computer Interaction That Reaches Beyond Desktop Applications, In Proc. of 42nd SIGCSE Technical Symposium on Computer Science Education, (Dallas, TX, March 9-12, 2011), ACM, 595-600.

[10] Malizia and Bellucci, A., The artificiality of Natural User Interfaces; Communications of the ACM, vol. 55, no. 3 (March 2012), ACM, 36-38.

[11] Muppala, J., 2010. Teaching Embedded Software Concepts Using Android, In Proc. of the 6th Workshop on Embedded Systems Education, (Tapei, Taiwan, October 09 - 14, 2011), ACM, 32-37.

[12] Norman, D. A., 2010. Natural User Interfaces Are Not Natural, Interactions (May + June 2010), ACM, 6-10.

[13] Norman, D. and Nielsen, J., 2010. Gestural Interfaces: A Step Backward in Usability, Interactions (September + October 2010), ACM, 46-49.

[14] Pauca, V.P. and Guy, R.T., 2012. Mobile Apps for the Greater Good: A Socially Relevant Approach to Software Engineering. . In Proc. of 43rd SIGCSE Technical Symposium on Computer Science Education, (Raleigh, NC, February 29–March 3, 2012), ACM, 535-540.

[15] Riley, D., 2012. Using Mobile Phone Programming to Teach Java and Advanced Programming to Computer Scientists. In Proc. of the 43rd ACM Technical Symposium on Computer Science Education, (Raleigh, NC, February 29–March 3, 2012), ACM, 541-546.

[16] Wigdor, D. and Wixon, D., 2011. Brave NUI World: Designing Natural User Interfaces for Touch and Gesture, Morgan Kaufmann, Burlington, MA.

Behavior Mining for Big Data: Promoting Critical Thinking in Data Science Education

Zhengxin Chen
Department of Computer Science
University of Nebraska at Omaha
Omaha, NE 68182-0500
zchen@unomaha.edu

Abstract—**The concept of behavior mining has been proposed some time ago. Instead of focusing on the data alone, behavior mining endorses the idea of exploring and mining the underlying natural, social and other kinds of behaviors which have produced the data, and discovering hidden interesting patterns behind these behaviors. Behavior mining for big data offers excellent opportunity for promoting critical thinking in data science education. In this paper we discuss the importance of behavior mining at the era of big data, describe implications of behavior mining for big data, and outline three case studies to illustrate its applications, and to show how behavior mining can promote critical thinking in data science education.**

Keywords—Big Data, data mining, behavior Mining, critical thinking, data science education

I. Introduction

The concept of behavior mining proposed some time ago [1], is aimed at a more in-depth understanding of what is behind the data, i.e., the *behavior* which has produced the data. Although behavior mining has not attracted a lot of attention, it makes strong sense at the era of big data. In fact, by offering an effective way to examine the data in a holistic and dynamic manner, behavior mining promotes critical thinking, and thus can play an important role in data science education. In this paper, we provide an overview on this perspective. Starting with a review of the basic concept of behavior mining (Section 2), we discuss the importance of behavior mining on big data (Section 3). Since the requirements of behavior mining could be quite demanding, a discussion follows suit on behavior mining of big data through critical thinking (Section 4). We also provide three case studies to illustrate applications of behavior mining of big data through critical thinking (Section 5). We conclude our paper in Section 6.

II. From Data Mining to Behavior Mining

As proposed quite a few years ago, in-depth data mining requires the move to behavior mining [1]. In fact, the term data mining is a misnomer, because this term suggests mine the data itself, which is apparently misleading. Data is just numerical numbers or character strings – if we don't consider the underline meaning of the data. Instead of focusing on the data alone, behavior mining endorses the idea of exploring and mining the underlying natural, social and other kinds of behaviors which have produced the data, and discovering hidden interesting patterns behind these behaviors. Note that the word *behavior* as appeared in the term behavior mining has a broad-sense meaning: Although it may refer to human (individual or social) behavior, it may also refer to any phenomenon produced due to biological, social, physical, or any other processes. Therefore, behavior mining is devoted to the study of the very nature of things or events we are interested. An intuitive example is the famous Brownian motion, which is the random motion of particles suspended in a fluid resulting from their collision with the quick atoms or molecules in the gas or liquid. In the case of Brownian notion, if we limit our attention only to the surface data (i.e., the geographical locations of the particles) alone, then "mining" such kind of data is completely meaningless. Similarly, focusing on the given, static data alone may not be able to reveal the real hidden knowledge behind the data – *unless the underlying processes which have produced the data are also considered together*. This is the essence of behavior mining.

III. Behavior Mining Meets Big Data

Although no standard definition yet, the term big data usually refers to data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it [4]. Big data is usually characterized by four V's (volume, velocity, variety and veracity) [3]:

- *Volume:* Not only can each data source contain a huge volume of data, but also the number of data sources has grown to be in the millions. Even for a single domain, this number is now in the tens of thousands. This is much higher than the number of data sources considered in traditional data integration.

- *Velocity:* Many of the data sources are very dynamic, and thers number of data sources is also exploding.
- *Variety:* Data sources (even in the same domain) are extremely heterogeneous both at the schema level regarding how they structure their data and at the instance level regarding how they describe the same real world entity, exhibiting considerable variety even for substantially similar entities.
- *Veracity:* Data sources (even in the same domain) are of widely differing qualities, with significant differences in the coverage, accuracy and timeliness of data provided.
- Big data implies a dynamic, ever-changing environment, which always requires integration.

Due to these unconventional features (comparing to those data handled by traditional database management systems or information retrieval systems), big data imposes huge challenges and offers tremendous opportunities as well. By focusing on the underlying behavior of the data, behavior mining should play a critical role in analysis of big data.

Big data makes behavior mining *necessary*, because when we analyze any form of big data, if we stay with the traditional practice of data mining, we will be drawn to huge numbers and character strings. Big data also makes behavior mining technically *possible*. This is due to the wide availability of data through the open environment, elasticity and dynamicity of the data, as well as many other factors. As such, the concept of behavior mining should be applied in computer science education, because it promotes critical thinking (to be elaborated a little later).

There are profound implications of behavior mining for big data. Some quick examples are:

(a) Identify (manually, semi-automatically, or automatically) and acquire additional data (beyond the given data) or other sources for analysis and mining.

(b) Unlike traditional data mining which is usually treated as a standalone process, behavior mining can be naturally incorporated into many applications. This is because behavior mining sets the emphasis on the dynamic process of the behavior (i.e., the application itself), rather than the static resulting data produced by this process.

Data science is the study of the generalizable extraction of knowledgeable from data. It incorporates varying elements and builds on techniques and theories from many fields. Although data Science is not only for big data, the fact that data is scaling up makes big data an important aspect of data science [9]. Since behavior mining endorses a holistic view of understanding big data by looking at the data from its origin, its progression, and its dynamicity, it acts as an appropriate discipline for studying data science. And this new discipline needs an effective way of carrying out all the needed tasks (such as identifying relevant resources, capturing dynamic aspects of various applications, etc.). Critical thinking is an important step to fill this gap.

IV. Behavior Mining of Big Data through Critical Thinking

As the study of clear and unclear thinking, critical thinking can be defined as the intellectually disciplined process of actively and skillfully conceptualizing, applying, analyzing, synthesizing, and/or evaluating information gathered from, or generated by, observation, experience, reflection, reasoning, or communication, as a guide to belief and action [10]. Critical thinking involves problem solving, decision making, metacognition (aka cognition about cognition), rationality, rational thinking, reasoning, knowledge, intelligence and also a moral component such as reflective thinking. Critical thinkers therefore need to have reached a level of maturity in their development, possess a certain attitude as well as a set of taught skills.

Critical thinking offers an effective way of behavior mining for big data, because critical thinking calls for a number of abilities for managing, studying and understanding big data, which include:

- *Recognize problems,* to find workable means for meeting those problems – Unlike "classical" data mining which only looks at the given static data, behavior mining requires us to study the data in its entirety, and in a progressive way (i.e., how data is produced and evolved). Therefore, behavior mining requires identifying the underlying problem beyond the given data. In other words, behavior mining is aimed not only at finding hidden knowledge patterns, but the *hidden problem* at the first stage.

- *Gather and marshal pertinent (relevant) information* – in the case of behavior mining for big data, we have to gather additional data and information (along with other evidence) beyond the given data to solve the identified hidden problem (as discussed above). Ideally, this step is preferably automated, but usually requires human intervention, due to its complexity. That is an important aspect of critical thinking.

- *Interpret data and recognize the existence (or non-existence) of logical relationships between propositions:* Although various techniques of artificial intelligence (AI) and related field should be applied whenever possible, human cognition is required at first hand to the logistics behind complexity of real world phenomena, so that we will be able to understand the role of each piece of required data and how these pieces are interrelated to each other.

As noted by researchers in critical thinking community, one of the Elements of Thought (which is a checklist for reasoning to improve their thinking in any discipline or subject area) [6] states that all reasoning is based on data, information, and evidence**.** This implies the need of searching for information that opposes one's position, as well as information that supports it. These elements require that we have to make sure that we have gathered sufficient information. There are also several universal intellectual standards involved in critical thinking for big data mining, including clarity, relevance, depth (which is concerned with how to address the complexities in the question) and breadth (which is concerned with alternative point of views). Put it all together, critical thinking offers useful hindsight and effective principles to face the challenges of volume, velocity, variety and veracity of big data, and should serve as foundation of today's data science education.

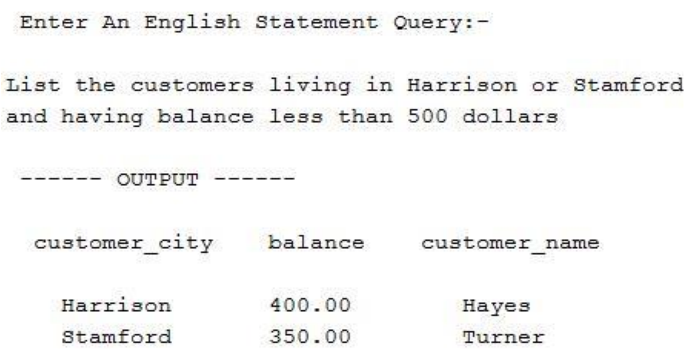
V. Case Studies

An examination of teaching and research of big data at upper undergraduate and graduate levels offered in our institution, and joint research conducted between our research lab with local business/ industry have revealed the potential of behavior mining in provoking critical thinking in data science education. Below are three case studies to show various aspects of conducting behavior mining through critical thinking.

A. Case study 1: Keyword search for streamlining data

Database keyword search has achieved much attention in the last two decades. Database keyword search not only relieves users' burden of writing SQL queries, but also offers a unified approach of accessing different formats of textual data (structured, unstructured and semistructured data), as well as facilitates database exploration. Driven by behavior mining, we can conduct database keyword search from a unique perspective. The motivation of the particular approach we have taken is based on the following important observation, namely, the complexity involved in generating operator trees or joint networks. A number of research papers on keyword search on data streams have offered various ways trying to tackle this issue, yet they still have to consider various operator trees or connected trees (e.g., [4,7]). In contrast to these approaches, we use an *alternative* way of database keyword search using critical thinking (through one of the universal intellectual standards mentioned above, namely, *breadth*): Instead of using individual keywords alone, we consider *how to interpret users' information needs behind individual keywords they have provided.* We have built a prototype system which stays with the simplicity of keyword search, yet it can incorporate the contextual information provided in the user query [8]. Unlike other approaches of keyword search where users enter individual terms (i.e., keywords), we allow users enter simple English sentences. Instead of resorting on a full-fledged natural language processing to process these queries in English sentences, the unneeded words in the queries are discarded. Only the specific contextual information along with the keywords containing database contents will be used to construct SQL queries. The contextual information is used to interpret the meaning of the queries, including the semantics involving AND, OR and NOT. At the end of this process, a single SQL query will be generated. A sample expanded keyword query "List the customers living in Harrison or Stamford and having balance less than 500 dollars" in a banking database produced by our prototype system is shown in Fig. 1 (taken from [8]).

Fig. 1. Screen shot for sample query



```
 Enter An English Statement Query:-

List the customers living in Harrison or Stamford
and having balance less than 500 dollars

 ------ OUTPUT ------

  customer_city    balance    customer_name

    Harrison       400.00        Hayes
    Stamford       350.00        Turner
```

Our original approach developed for static databases [8] is adopted to handle continuous streaming data which is captured within a time window. The window is maintained in a way so that oldest data is removed while new data is added. Since only one SQL query will be generated, result is generated in a real-time fashion. In addition, this approach has the potential of tracking the concept drift of data over time.

B. Case study 2: Mining knowledge from clickstream data

Clickstream data is an example of applications involving data streams. We have conducted a joint research with local industry on such data related to travel-related online activities. The travel advertising firm wants to focus on online users looking out for car rental, hotel booking, tourism etc. The objective of this project is to come up with a pool of promising users who have higher probability of clicking ads on their browsers, which may lead to a purchase. As a result, there is a need to analyze and predict travelers' behaviors. The company can then popup more ads (impressions) on the browser of such cookie-ids, and gain profit from each user click. Each user or cookie-id, will be given a numerical score. The score indicates how likely a user is going to click the ads on his /her browser to make a purchase; in general, the higher the score, the more is the likeliness of that user to click the ads. By doing so, we can filter users as per their proclivity towards ads. This score also indicates how much money is worth investing on the user. The higher the score, more money can be invested on advertising such users since they have higher potential of clicking the ads. Main steps of this project include using training data to build a classification model, using testing data to check the accuracy of the model, and applying the model to the actual data set. Typical classification algorithms such as Decision Tree and SVM regression have been used to implement our approach.

An interesting aspect of this project is that it illustrates the necessity of dynamic incorporation of additional data – as required by behavior mining and stimulated from critical thinking. For example, there is a need for handling airport information appropriately, because travel-related activities such car rental or hotel booking all depend on it. One may think this is just an issue related to geographical information, but actually it is much more than that. For example, consider the case of New York City area. There are three major airports in this area, and they belong to different states. Other major cities in the other countries throughout the world such as London, Paris and Tokyo also have multiple airports, and each country is different from other countries. Even within the same country, NYC is very different from southern California (such as LAX and Burbank in Los Angeles area), and business rules related to hotel booking and car rental could be very different. How can we identify relevant resources and automatically incorporate them into dynamic clickstream data presents a major challenge. In addition, as the data size grows, the diversity of data also grows, which triggers the inclusion of even more data, such as economic development of various districts or counties, as well as censor's data related to population, etc. A completely different direction of expansion is on the user's side: whenever applicable, appropriate knowledge involving profiles of users can also be incorporated. This results in a big "closure," or a data-ecosystem which may reveal hidden knowledge involving various social-economic aspects behind clickstream data. As the result, more targeted ads can be produced.

C. Case study 3: Recommender system

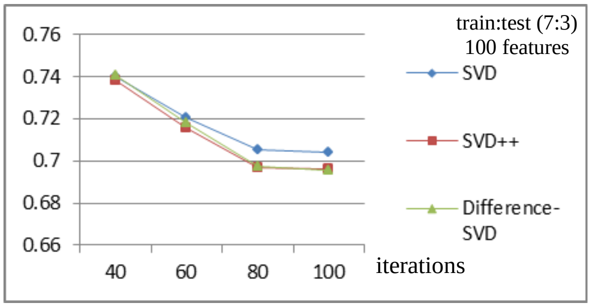
Although behavior mining is intended to analyze all kinds of behaviors (thus is not restricted to narrow sense of behavior, such as customer behavior), the recent study of social behavior in social networks undoubtedly offers strong motivation in fueling this research. We have just conducted a study on recommender systems to predict the 'rating' or 'preference' that user would give to an item based upon historical grading datasets. Existing approaches such as Singular Value Decomposition (SVD) has shown promising results. However, even with improvement made by SVD++, there are numerous ways to provide implicit feedback. Information gained by overlapping of communities with clusters is the motivation of our proposed method, which is resulted from our critical thinking (addressing the complexity of the question under investigation -- the *depth* of the Universal Intellectual Standard). Our work makes use of Louvain-method for identifying communities based on users' association in the network; it also uses k-means to cluster users based on their preferences. Communities indirectly represent the friend circle for users, whereas clusters represent users having similar taste in the recommended items (such as movies). The results of the above two methods are combined to get user groups with finer granularity having similar profiles. The resulting model is referred to as *difference-matrix model*, because it makes use of the proposed *difference-matrix* to associate a user's community and cluster information [2]. A sample difference matrix for a movie recommender system is shown in Fig. 2. A comparative study of experiments involving our proposed model is shown is shown in Fig.3, where the X-axis is representing number of iterations used for parameter training, and the Y-axis represents *root-mean-square deviation* (RMSE) of the testing dataset. (Both Figures are taken from [2].)

Fig. 2. A sample difference matrix

Movies Difference to Movie 1

$$\begin{array}{c|ccccc}
\textbf{Users} & \textbf{1} & \textbf{2} & \textbf{3} & \textbf{4} & \textbf{5} \\
\hline
\textbf{1} & 0 & -1.5 & 0 & 2 & 1 \\
\textbf{2} & 0 & -1.5 & 0 & 2 & 1 \\
\textbf{3} & 0 & -1.5 & 0 & 2 & 1 \\
\textbf{4} & 0 & -1.5 & 0 & 2 & 1 \\
\textbf{5} & 0 & -1.5 & 0 & 2 & 1 \\
\textbf{6} & 0 & 1 & 1 & none & none \\
\textbf{7} & 0 & 1 & 1 & none & none \\
\textbf{8} & none & 1 & 1 & none & none \\
\textbf{9} & 0 & 1 & 1 & 0 & none \\
\textbf{10} & 0 & 1 & 1 & 0 & none
\end{array},$$

Fig. 3. Result of a comparative study shows the effect of proposed Difference SVD model



VI. Conclusion

In this paper we reviewed the concept of behavior mining and discussed its implication with big data analysis. We have discussed how behavior mining can promote critical thinking in data science education. Case studies have also been provided to analyze the use of behavior mining for critical thinking in problem solving related to big data. Although behavior mining is particularly suitable for handling big data and conducting big data analysis, and critical thinking has shown impressive potential for conducting the research, the study is still at beginning stage. When we are facing any real world problem related to big data, we should *purposely* apply behavior mining at the very beginning of the problem solving process. A full range of research agenda for behavior mining in data science education is yet to be developed.

ACKNOWLEDGMENTS

Case studies presented in this paper have been conducted by members of Data Science Lab at College of Information Science and Technology (IST), University of Nebraska at Omaha (Prof Y. Shi and the author are the co-director of the lab). Supports of the Data Science Lab from the Dean's office of the IST College are highly appreciated. Several graduate students, particularly R. Patil and W. Deng, played key roles in case studies. Figures included in this paper are taken from the related publications of which the author is a co-author. However, the author is solely responsible for the theoretical perspective presented in this paper.

REFERENCES

[1] Z. Chen, From Data Mining to Behavior Mining. International Journal of Information Technology and Decision Making 5(4): 703-712 (2006)

[2] W. Deng, R. Patil, L. Najjar, Yong Shi, Z. Chen, Incorporating Community Detection and Clustering Techniques into Collaborative Filtering Model, Procedia Computer Science, Proc. Information Technology and Quantitative Management (ITQM2014)

[3] . L. X. Dong and D.Srivastava, Big Data Integration (tutorial summary), VLDB 2013, 1188-1189.

[4] M. Eltabakh, Introduction & Logistics for CS525: Special Topics in DBs: Large-Scale Data Management, Spring 2013, WPI. (Online slides)

[5] A. Markowetz, Y. Yang and D. Papadias, Keyword search on relational data streams, Proc. SIGMOD 2007.

[6] The National Council for Excellence in Critical Thinking, A Draft Statement of Principles (http://www.criticalthinking.org/pages/the-national-council-for-excellence-in-critical-thinking/406)

[7] L. Qin, J. X. Yu, L. Chang and Y. Tao, Scalable keyword search on large data streams, Scalable keyword search on large data streams. *VLDB J.2011*, 20(1), 35–57.

[8] R. Patil, Z. Chen: STRUCT: incorporating contextual information for English query search on relational databases. KEYS 2012: 11-22

[9] Wikipedia, Critical thinking, http://en.wikipedia.org/wiki/Critical_thinking

[10] Wikipedia, Data Science, http://en.wikipedia.org/wiki/Data_ience

Interdisciplinary Team-Teaching Experiences of a Computer and Nuclear Energy course designed for Non-Nuclear Engineering Students

Peter Keiller, Charles Kim, and Robert Rwebangira
Howard University, Washington DC, U.S.A.

Abstract

This paper discusses an experience and assessment of an interdisciplinary computer and nuclear safety course which was developed and offered to non-nuclear engineering students. Computer control and nuclear energy safety were the two areas addressed. A unique interdisciplinary team-teaching model was applied to teach the two areas. The university professors taught the sections dealing with computer dependable systems with emphasis placed on reliability and safety while the guest speakers focused on topics related to nuclear energy safety and security. This teaching model allowed an institution without a nuclear engineering program to offer nuclear-related courses. The overall survey results from the offerings of the course showed a very positive response from the majority of the students.

Keywords - interdisciplinary course; nuclear safety; computer related accidents; team teaching; hardware/software diversity

1 Introduction

The rapid adoption of computer control systems for nuclear power plants has resulted in the reliability of computer hardware and software becoming an important component of nuclear plant design, operation, and maintenance. Security and reliability of hardware and software plays a crucial role in the safety of nuclear plants. The need for safe and reliable computer based control systems has created new and greater concerns that were not experienced during the age of analog control systems. Because of these concerns, the failure of a single component, such as a defected control logic device or faulty coding procedures or subroutines, could disable major functions of the system and cripple the successful operation of any nuclear plant. This critical situation requires a timely supply of capable engineers who can deal with the new issues of the safety and reliability of computer control systems in nuclear plant operations. There are numerous practical challenges related to teaching nuclear engineering and science courses to non-nuclear students, especially in an institution with no programs related to nuclear engineering.

To address the shortage of new engineers in the emerging issues of computer control and to provide a broad interdisciplinary knowledge to general engineering students, a new teaching model was needed to develop "a more comprehensive suite of mechanisms that can be implemented to diversify and to add excitement to the nuclear field" [1]. A new course development project was proposed to the U. S. Nuclear Regulatory Commission (NRC) with aims to diversify and to add excitement for non-nuclear engineering students to the nuclear engineering field [2]. The project was granted and a team-teaching model was adopted and organized, comprising of the project investigators teaching computer safety methodologies and the nuclear experts focusing on areas dealing with nuclear engineering safety. The new course was developed and offered with the team-teaching model. The NRC guest speaker series formed with volunteer nuclear experts with diverse backgrounds greatly helped to develop the interdisciplinary nuclear course.

2 Course Contents Development

The team-teaching based new course was developed with emphasis on organization with defense-in-depth concepts; emphasis on hardware/software diversity relevant to digital instrumentation and control (I&C) and regulations, cyber-security, and safety within nuclear energy organizations.

2.1 Course Contents for Computer Safety

Digital Instrumentation and Control: A nuclear power plant has a central control room where operators collect, detect, analyze, monitor, and verify information from multiple indicators and alarms. The majority of I&C systems in today's nuclear plants are beginning to apply advanced I&C technology in all aspects of operation and maintenance.

Software Errors: Software used in nuclear plants must be ensured that required actions are taken to avoid unnecessary fixes. A number of software errors have been detected in the operation of nuclear plants. It has been documented that the failures due to software errors occurred as often as those derived by hardware failures. The major concern with software errors are that these errors tend to be difficult to prevent because they may occur only when an unusual set of inputs exist.

Computer Technology and Common Mode Failure: Computer control system faces safety issues in that common mode failures may fail even redundant safety systems compromising safety detecting functions [3]. Since software based I&C is known to be vulnerable especially to common-mode failure, the system requires deploying different diversification strategies in the applications with also redundancy in the hardware and operating systems.

Defense-in-depth and Diversity: All safety activities are subject to layers of overlapping provisions so that if a failure occurred it would be compensated for or corrected without causing harm to individuals or the public at large. The defense-in-depth concept, when properly applied, ensures that no single human or mechanical failure could lead to injury of the public and that even combinations of failures which are only remotely possible would lead to little or no injury. In the design diversity approach, multiple versions of software of different algorithms can be written for the same function. For hardware, processors and operating systems from completely different architectures and designs can be employed for the same function [4].

Hardware Diversity Kit: The hardware diversity kit is composed of a nuclear reactor event scenario generator and a set of diverse architectural hardware which individually responds to the scenarios [5]. For coding each of the platforms, students will be required to use different behavioral requirements for the same function for the scenario.

2.2 Course Contents for Nuclear Energy

NRC Overview: This topic covers the mission and the major activities of NRC in the areas of material, reactors, oversight, emergency responses, and new researches.

Nuclear Reactor Concepts: Commercial nuclear power plant design categories are discussed covering pressurized water reactors and boiling water reactors.

Security and Safeguard of Nuclear Power Plant: Nuclear power plants must be both safe and secure. Safety refers to operating the plant in a manner that protects the public and the environment. On the other hand security refers to protecting the plant by using trained staff, equipment and other methods from intruders who wish to do harm.

Licensing Process: The steps and the level of details regarding the licensing are discussed for stable and predictable licensing process so that safety and environmental issues are resolved before authorizing construction. The involvement of timely and meaningful public participation is provided.

Power Generation: Design considerations and regulations are discussed for technical specifications and environmental qualification as well as station blackout and maintenance rules. Worldwide industry trend is also presented.

Digital System Safety and Cyber-Security: This is to ensure digital safety system reliability, availability, and integrity for non-malicious and malicious events. The topic talks about the history of digital system safety and cyber-security; overview of the current cyber- security program and digital system safety review; and the regulatory developments regarding cyber security.

Severe Accident Analysis: This topic compares the design basis accidents and severe accidents in terms of accident severity as well as the station blackout and plant response.

New Rectors: This topic discusses the new and small modular reactors with specific features and safety systems.

3 First Course Offering

The new course, officially numbered and titled as “EECE499 Computers and Nuclear Energy” at Howard University, was offered for the first time in fall 2011 for all engineering students of both undergraduate and graduate status. The course was subsequently offered in the fall 2012 and 2013.

3.1 Course Offering Format

The course was conducted with a weekly schedule covering the two subjects (Computer control and nuclear energy safety) simultaneously. Further details of the class, including the materials covered, are available from the website of the project at http://www.mwftr.com/NuclearSafety.html.

3.2 Survey and Assessment

For the 2011, class a total of 26 engineering students, 19 undergraduate (of electrical, computer, and chemical engineering) and 7 graduate (of electrical engineering), out of 28 students enrolled in the class completed a class survey and indicated their perception on the course. Overall, the students expressed very positive attitudes in taking the course. In particular, the students were very satisfied with the format of the interdisciplinary team-teaching and the team of lecturers – their professor on computers safety and the guest speakers on nuclear energy. Overall, they rated the course quite favorably in terms of their (a)understanding in computer mistakes and errors (b) understanding nuclear energy and safety, (c) understanding the use of computer in nuclear area, (d) increase in their interests in safety-critical computer systems, and (e) meeting their expectation of the course.

Tables (1-3) show (a) Lectures presented by the NRC and HU faculty members, (b) the overall average survey of the course (1-5 with 5 being the highest score), and (c) the students wish list for future offerings

Table 3 shows the Students Wish-List (Top 5). This List is being addressed in future offerings of the course.

4 CONCLUSIONS

This paper described an NRC sponsored project of teaching computer safety for digital I&C in nuclear plants for non-nuclear engineering students at Howard University by applying a team-teaching model to effectively cover two subjects in safety concerns: computer control and nuclear energy. The project investigators and the NRC guest speakers covered the two subjects concurrently for the developed course. The majority of students of the course agreed that the course met their expectation and increased their knowledge and understanding of the subjects. This project, when further enhanced, would meet the demand not only for training students but also eventually for diversifying workforce in computer and nuclear safety.

Table 1
EECE499 Computers and Nuclear Energy - Lecture Presentations (NRC/Faculty)

a) Computers and Risk (Faculty)
b) NRC Overview (NRC)
c) Introduction to Nuclear Reactor Concepts (NRC)
d) Security and Safeguards of NPP (NRC)
e) Software Reliability in Safety Critical Systems (Faculty)
f) Software Failure Data Analysis (Faculty)
g) Nuclear Criticality and Nuclear Engineering (NRC)
h) Defense in depth (Faculty)
i) Power Updates (NRC)
j) Diversity for Safety Critical Systems Applications (Faculty)
k) Licensing Processes (1) and (2) (NRC)
l) IEC Standards of Safety – Critical Systems (Faculty)
m) Cyber-security (NRC)

Table 2
EECE499 Computers and Nuclear Energy - Student Survey (1-5 (5 being the highest score))
Part I

a) Students understanding of computer errors/failures (4.89)
b) Students understanding of nuclear energy and safety (4.56)
c) Students understanding of the use of computers in the nuclear industry (4.44)
d) Students interested in safety critical systems (4.56)
e) Students interested in the nuclear energy and safety (4.00)
f) Students interested in employment in the nuclear energy field (3.89)
g) Students expectation of the course (4.78)

Part II

a) Content and presentation on safe computing (5.00)
b) Adequacy of assignment on computers (4.89)
c) Amount of knowledge gained on safe computing (4.67)
d) Content and presentation on nuclear energy (4.67)
e) Knowledge gained on nuclear energy (4.56)
f) Level of comfort in class lectured by team teachers (4.89)

Table 3
EECE499 Computers and Nuclear Energy - Student Wish List

a) More Hands-on activities
b) More on-site visits – especially to nuclear power plants
c) More utilization of Multimedia CDs
d) More software coding examples
e) Earlier involvement during the semester with the Hardware Diversity Kit

5 Acknowledgment

The authors thank all the guest speakers who lectured for the Computers and Nuclear Energy courses in the fall semesters, 2011-2013, at Howard University, Washington , DC.

6 References

[1] "The Future of University Nuclear Engineering Programs and University Research & Training Reactors" Summary record of hearing before Committee on Science, United States House of Representatives (2004)

[2] C.Kim and P.Keiller, "A Course Development Project for Hardware Diversity and Software Reliability Education for Digital Instrumentation and Control of Nuclear Power Plant ," 2011 Conference on Nuclear Training and Education, Feb 6-9. 2011, Jacksonville, FL.

[3] "Digital Instrumentation and Control Systems in Nuclear Power Plants-Safety and Reliability Issues", National Research Council, National Academic Press, 1997, Washington, DC.

[4] F. Salewski, D. Wilkin, and S. Kowaleswski, "Diverse Hardware Platforms in Embedded Systems Lab Courses: A Way to Teach the Differences," ACM SIGBED Review, 2 (4) , pp. 70-74, (2005).

[5] O. Sonoiki and C. Kim, "Development of a Hardware Diversity Training Kit for Nuclear Industry", ANS 2011 Student Conference, April 14-17, 2011. Atlanta, GA.

Attachment-Theory Based Pedagogy

Karina Assiter
Department of Computer Science and Networking
Wentworth Institute of Technology
Boston, MA

Abstract
There are a plethora of formal methods for educating undergraduate students, including recommendations for classroom management, concept presentation and assignment creation. What is often overlooked is the influence that student-instructor attachment might have on performance. On the primary and secondary levels there is evidence to support what we would assume to be true; that attachment does effect performance. We then assume that, to some degree, these findings would carry over to college-based instruction. Instructors have their preferred teaching styles (based on their personality types and learning styles), but regardless of the variations that exist and how well they correspond to student preferences, what is essential to learning is student motivation. Motivation could be influenced by a combination of factors, including inherent interest, career objectives, external pressures (i.e., parental expectation), and appropriate (based on skill level) complexity of assignments. This paper proposes that student motivation, and, therefore, performance, are also (on the secondary level) affected by student-instructor attachments. First, we discuss the results of attachment-theory research with K-12 students. Then we propose a (first) case where, on the secondary level, the student-teacher attachment should be explicitly (versus implicitly) developed/improved. Finally, we start to investigate methods that could eventually be formally evaluated for improving (healthy) student-teacher attachments, and, therefore, student performance, at the secondary level.

Keywords: Attachment-theory, Person-Centered teaching and learning,

1 Attachment in K-12

It's understood in K-12 education that the relationship between the teacher and the student impacts the degree and quality of learning. Dr. Marvin Marshall, in his article entitled "Learning and Relationships: The two are inseparable" [1] provided an illustrative scenario where, when students gave a teacher trouble, the response of the students responsible was that "the teacher did not like them". He claims that "when students harbor a visceral dislike, the teaching and learning suffer". Marshall recommends that teachers "encourage, empower, find an interest and build on it, be positive, develop procedures to help students help themselves". "Superior teachers", he states, "avoid stimulating students to have negative feelings toward them".

Peters & Le Cornu [2] argue for *constructivism* as a theoretical basis for student learning. The idea is that "the learner is active in the process of taking information and building knowledge and understanding". As they state, "central to constructivism is the notion that learners play an active role in constructing their own meaning." They argue for the "teachers moving away from the teacher centered conversations to ways of interacting with individual learners in … relation to a real problem they are solving". In studying several classrooms that applied the constructivist approach, the teachers in the study expressed the belief that the relationships that they have with each student (sic) are integral to a classroom culture that optimizes learning." The role of the instructor is to help each student develop self-directed learning skills as well as confidence in their own abilities. Of course, the authors acknowledge that "monitoring each student's progress and needs is problematic."

As the authors acknowledge in [2] "the relationship between teachers and students is a neglected area of computing education research." These authors cite several studies that link student's attitude toward the lectures/class with a positive relationship with the lecturer. They quoted one lecturer as saying, "if you want to be a good teacher, you really have to show the students . . . that you are passionate about the things you are teaching. The students can very quickly discover the fraud, so you must actually show your love of the material, if that comes across I think half the battle is won."

Riley [3] proposed that classroom dynamics are influenced by the attachment histories of not only the students, but also that of the teachers. In terms of the use of the word "attachment", the authors' state that "Attachment is the most comprehensive theory describing human relationships." The traditional assumption was that the teacher, as the "grown up." does not misbehave (they're professional and neutral), and if they do they are a "bad teacher"; the student, on the other hand, may misbehave due to their personal histories and/or current issues. Instead, Attachment-Theory proposes that attachment is between individuals who each bring their own (often unconscious) assumptions and behaviors to the dynamic. In Riley's study, when teachers were able to recognize and understand their own attachment patterns then they were able to change negative classroom behaviors. Though formal Attachment Theory research has, thus far, only focused on K-12 education, we expect that unhealthy attachment-based behaviors exist in secondary education.

2 Attachment in Secondary education

Our objective in writing this initial concept paper on the role of attachment in secondary education is to:

a) Suggest a broadening of focus when we discuss pedagogical improvements; in addition to instructional methods, we should also suggest that instructors be mindful of their own (and students) potential attachment-based behaviors.
b) Propose future research that will provide a theoretical foundation to what we suggest to be true based on anecdotal evidence.
c) Solicit investigators who are interested in exploring this topic

As the Psychology researcher and author Bryne Brown highlights in her TED talk [3] what really matters to us as humans is connection with others; so either having or not having that connection with a student will impact overall influence and therefore, learning.

2.1 When Attachment Matters

In large research institutions student attachment with the primary instructor is neither likely nor valued by the institution, nor would it be expected from the students. Generally, the student would form attachments with the TA or the lab instructors; as these instructor stand-ins' are not first and foremost teachers (they are students, themselves), there is less expectation that they master the pedagogical tools, techniques and behaviors. To illustrate this point we consider the difference in expectations that a child has of its' parent versus the expectations that they would have of an older care-taking sibling.

At institutions (and within departments) where student retention is a concern, exceptional teaching is expected. When negative student-instructor attachment behaviors arise, the outcome could be poor student evaluations, decreased confidence on the part of the instructor and negative feelings toward students - which could then intensify the original behaviors. In the extreme case, this could lead to the replacement of the instructor; a loss of both the initial investment in their training, as well as the potential of having them as a successful contributor. This scenario may occur frequently; however, since attachment theory has not been addressed at the secondary level, we have no indication of its impact (financial, personal, departmental, etc.).

In research at the K-12 level, the humanist psychologist Carl Rogers [4] provided empirical support for the proposition that teacher-student relationships are fundamental to understanding classroom behavior by teachers as well as students and directly affects student outcomes. As one reviewer of the final unpublished writings of Rogers [4] states:

> *The principal finding of the research in this book show that teachers and schools can significantly improve their effectiveness through programs focusing on facilitative interpersonal relationships. Teachers who either naturally have, or are trained to have empathy, genuineness (congruence), and who prize their students (positive regard) create an important level of trust in the classroom and exert significant positive exert significant positive effects on student outcomes including achievement scores, interpersonal functioning, self-concept, attendance, and violence.*

We are finally realizing that other dimensions of pedagogy applied at the K-12 level have benefits in secondary education (active versus lecture-based learning); thus, we would expect that attachment theories could also improve teaching and learning in higher education.

2.1 Conditions for Attachments

The theories of person-centered teaching and learning [5] suggest that students learn best in an environment characterized by three attitudinal conditions (Table 1).

Table 1 - Attitudinal Conditions ideal for Person-Centered Pedagogy

Conditions	Synonyms
Congruence	Realness, genuineness, transparency, authenticity, openness
Acceptance	Respect, unconditional positive regard
Empathic Understanding	A deep understanding for the feelings and meanings of the other".

As Motschnig-Pitrik & Santos (2006) have pointed out, these attitudinal conditions "must be held and lived by the facilitators and communicated to the learners such that they actually perceive them and experience them as part of the teaching and learning relationship." Essentially, these conditions can give us a starting point for considering beneficial student-teacher attachments.

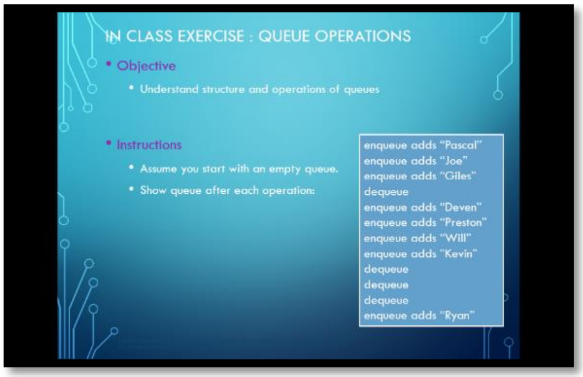
2.2 Cultivating Attachment

In the context of the traditional face-to-face lecture, educators with an extroverted personality type *may not need to explicitly* cultivate student-teacher attachments; they're natural openness would lead students to feel connected, even if the connection is only unidirectional. On the other hand, educators with an introverted personality type, who would be less open in the public forum of a lecture, *may need to explicitly* cultivate attachments with individual students'. These educators may choose methods such as those shown in Table 2.

Table 2 - Methods for Cultivating Attachments

Method	Condition(s) to communicate
Personalized feedback on student work	Congruence
Initiate conversation before or after class with queries such as: "so how are you doing?"	Congruence, Acceptance, Empathic understanding
Allow time in class for students to be able to meet with the instructor individually	Congruence, Acceptance
Assign project-based journal entries (opportunity for instructor to see and respond to personality and individuality of students)	Congruence, Acceptance
Include student names in examples/exercises on the lecture.	Congruence, Acceptance
Contact with individual students (email) demonstrating concern over students' performance (not in a reprimanding way but a concerned way).	Congruence, Acceptance, Empathic Understanding

Admittedly, cultivating beneficial attachments[1] with individual students is an inexact science; it's a matter of being watchful for opportunities. For example, in a class of very quiet students, an instructor added student names to a lecture-based exercise (Figure 1). Subsequent to the presentation of the lecture the students listed in the illustration appeared to be more engaged in class.



IN CLASS EXERCISE : QUEUE OPERATIONS

- Objective
  - Understand structure and operations of queues
- Instructions
  - Assume you start with an empty queue.
  - Show queue after each operation:

```
enqueue adds "Pascal"
enqueue adds "Joe"
enqueue adds "Giles"
dequeue
enqueue adds "Deven"
enqueue adds "Preston"
enqueue adds "Will"
enqueue adds "Kevin"
dequeue
dequeue
dequeue
enqueue adds "Ryan"
```

Figure 1 - Lecture with embedded student names

Certainly, the effect of these types of indirect student-teacher connections are hard to quantify; thus, we haven't yet tried to prove, definitively, that they enhanced student-teacher attachments (and therefore, learning). However, the anecdotal evidence from this example (and others like it) suggests that attachment-based pedagogy is an area worth exploring further.

3 Conclusions

In this paper we have suggested that at all educational levels (Higher-education as well as K-12):

1. Student motivation, and, therefore, performance, is affected by student-teacher attachments
2. Attachment-Theory based pedagogy[2] should be included in the development of all teachers.

In presenting our case, we first discussed the results of K-12 attachment-theory research that showed a strong correlation between student performance and instructor attachments. Then we discussed conditions for attachments based on person-centered teaching and learning research. Finally, we provided a (not at all comprehensive) list of example methods for cultivating student-teacher attachments.

4 Future Work

As this was a concept paper, the next challenging step is to demonstrate a definitive correlation between employed methods (to explicitly cultivate attachment) and student performance.

5 References

[1] M. Marshall, "Learning and Relationships: The two are Inseperable," 2003. [Online]. Available: http://www.Marvin.Marshall.com.

[2] G. Barker, "Making Visible the Behaviors that Influence Learning Environments: A Qualitative Exploration of Computer Science Classrooms," *Computer Science Education*, pp. 119 - 145, 2004.

[3] B. Brown, "Brene Brown : The Power of Vulnerability," June 2010. [Online]. Available: http://www.ted.com/talks/brene_brown_on_vulnerability.html.

[4] C. Rogers, H. C. Lyon and R. Tausch, On Become an Effective Teacher : Person centered Teaching, psychology, philosophy, and dialogues with Carl Rogers and Harold Lyon, Routledge, 2014.

[5] P. Riley, Attachment Theory and the Student-Teacher Relationship, New York, NY: Routledge, 2011.

[1] Attachments that promote learning.

[2] It can be argued that pedagogical methods, in general, should be included in the education of college professors (in addition to their particular fields of study).

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

[6] J. Peters and R. Le Cornu, "Constructing Relationships for Learning," in *NSARE/AARE Joint Conference*, *Auckland*, 2003.

[7] A. Berglund and R. Lister, "Introductory Programming and the Didactic Triangle," in *Proc. 12th Australasian Computing Education Conference (ACE 2010)*, Brisbane, Australia, 2010.

[8] J. East, "On Models of and for Teaching : Toward Theory-Based Computing Education," in *ICER*, Canterbury, United Kingdom, 2006.

[9] M. Hitchens and R. Lister, "A Focus Group Study of Student Attitudes to Lectures," in *Proc. Eleventh Australasian Computing Education (ACE) Conference* , Wellington, New Zealand, 2009.

[10] J. Cornelius-White, "Learner-Centered Teacher-student Relationships are effective : A meta-analysis," *Review of Educational Research,* pp. 113 - 143, 2007.

[11] R. Motschnig-Pitrik and A. M. Santos, "The Person Centered Approach to Teaching and Learning as Examplified in a Course in Organization Development," *ZFHE,* pp. 5 - 30, 2006.

Web-Based Database Project for the Lawton, OK Police Department: Interdisciplinary Systems Software Development at Cameron University

K. David Smith, Chao Zhao, Mike Estep, Abbas Johari
Computing & Technology Department, Cameron University, Lawton, OK, USA

William Carney
English Department, Cameron University, Lawton, OK, USA

Abstract – *Teaching systems software engineering to undergraduates can be a challenging task. Cameron University offers a unique interdisciplinary research and production experience for its senior Computer Science, Multimedia, Information Technology, and English majors. This capstone course provides students the opportunity to complete a comprehensive software project working for an actual client, and affords students the opportunity to practice real-life research and professional skills in a setting with multiple feedback opportunities. The client for the Spring 2013 course was the City of Lawton, OK, Police Department Gang Unit. The resulting student-developed product saved the city an estimated $100,000 in the cost of having to purchase a comparable product from industry. From this course, students learned to work as part of an interdisciplinary team, assess needs of the particular client, and use research skills to locate and utilize needed information. Additionally, student learning outcomes were enhanced through the use of a CASE tool.*

Keywords: Capstone Class, Software Engineering, Real-world Project, Interdisciplinary

1 Introduction

The Department of Computing and Technology at Cameron University offers three programs in the computing field: B.S. in Computer Science, B.S. in Information Technology, and B.S. in Multimedia Design. The Capstone is a required core course in these three programs. This course is designed to provide students with necessary skills, techniques, and tools to develop and manage complex software projects. In 2009, one of the authors brought the idea to the department of using an interdisciplinary approach to teach web-based software development classes, to include Computer Science (CS), Multimedia (MM), and Information Technology (IT) students. Since then, each semester's class has been designed to replicate an actual software development company. Recently, English (ENG) student technical writers have been added to these projects. The professors involved in the class act as the Chief Executive Council of the company. Potential projects for each semester are obtained through viral advertising. Once projects are requested, professors from each discipline look at the list of potential projects and choose the best with regards to required rigor and students' ability to achieve a working solution by the end of the semester. Only one project is chosen each semester. Once chosen, the professors meet with the potential clients and explain the clients' responsibilities, as well as how the class is managed. Major client responsibilities include providing an initial briefing to the students, attending two In Process Reviews (IPRs) in which students show their progress to the clients and get clarification for problems encountered, and attending the final presentation of the resulting product. Additionally, clients must provide letters of recommendation for the students involved as the only form of payment, as the students provide services on a pro-bono basis, in a service learning capacity. The client for the Spring 2013 course was the City of Lawton, OK, Police Department Gang Unit.

2 Initial Methods

2.1 Class Arrangement

After initial student team assignment and analysis, the IT and MM students would begin development of a backend database and aesthetic qualities of webpages. Upon backend database completion, focus of the project would shift to the CS students. The CS student emphasis would be on coding, testing, and validation of the project. MM students would continue to improve web page layout while IT students would work on alternatives [1, 2, 3], other documentation, and assisting with SQL code and changes to the backend as required. At this stage, the other teams would work in support of the CS team. ENG technical writers would edit documentation as necessary [1, 4]. Details of these procedures are provided in Section 2.2.

2.2 Organization and Teams

Teams were chosen through blind resumes and cover letters [1, 4]. The two established teams were charged with the developing their own solution to the client's problem. Having two separate solutions would provide an opportunity for the clients to select the end product that would best fit their needs. Technical writers from the ENG department were assigned to the teams upon integration. A typical integrated team consisted of a project leader, a lead programmer, a lead database designer, a lead webpage designer and a tech writer [4]. Other members were placed under these leads. An organization chart is shown in Figure 1.

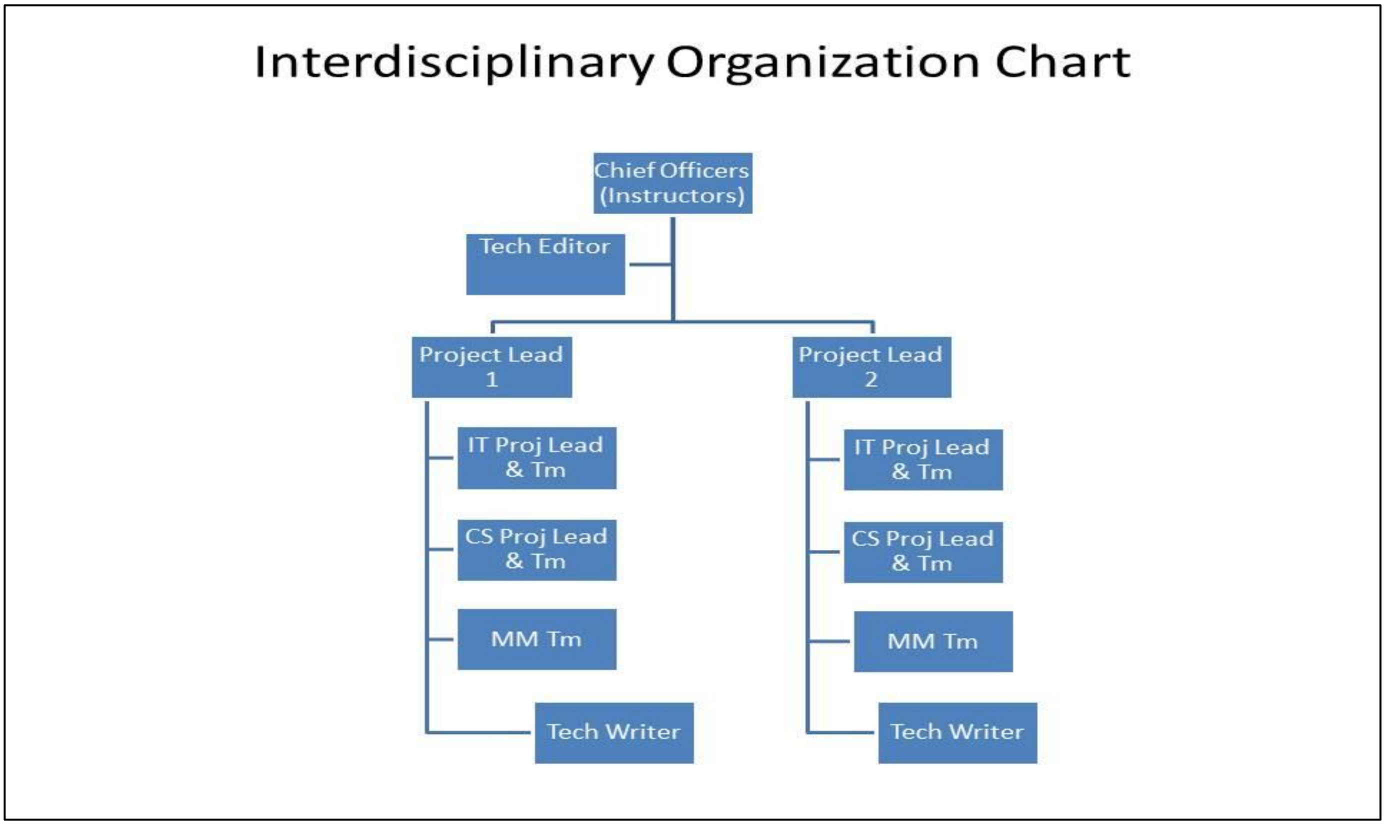


Figure 1. Organization Chart

2.3 Project Components

Development of the project components occurred in the following order:

(1) IT students developed a MySQL database through client interviews and analyzing the old client system.
(2) MM students began webpage development, and used HTML/CSS/JavaScript to code pages.
(3) CS students developed middleware using PHP to connect the frontend of the system to the backend database.
(4) CS students continued to refine webpage aesthetic qualities.
(5) Technical writers reviewed all the documents of the project and made all necessary corrections and editing.

An example of the inputs, processing, storage, and outputs can be seen in Figure 2.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

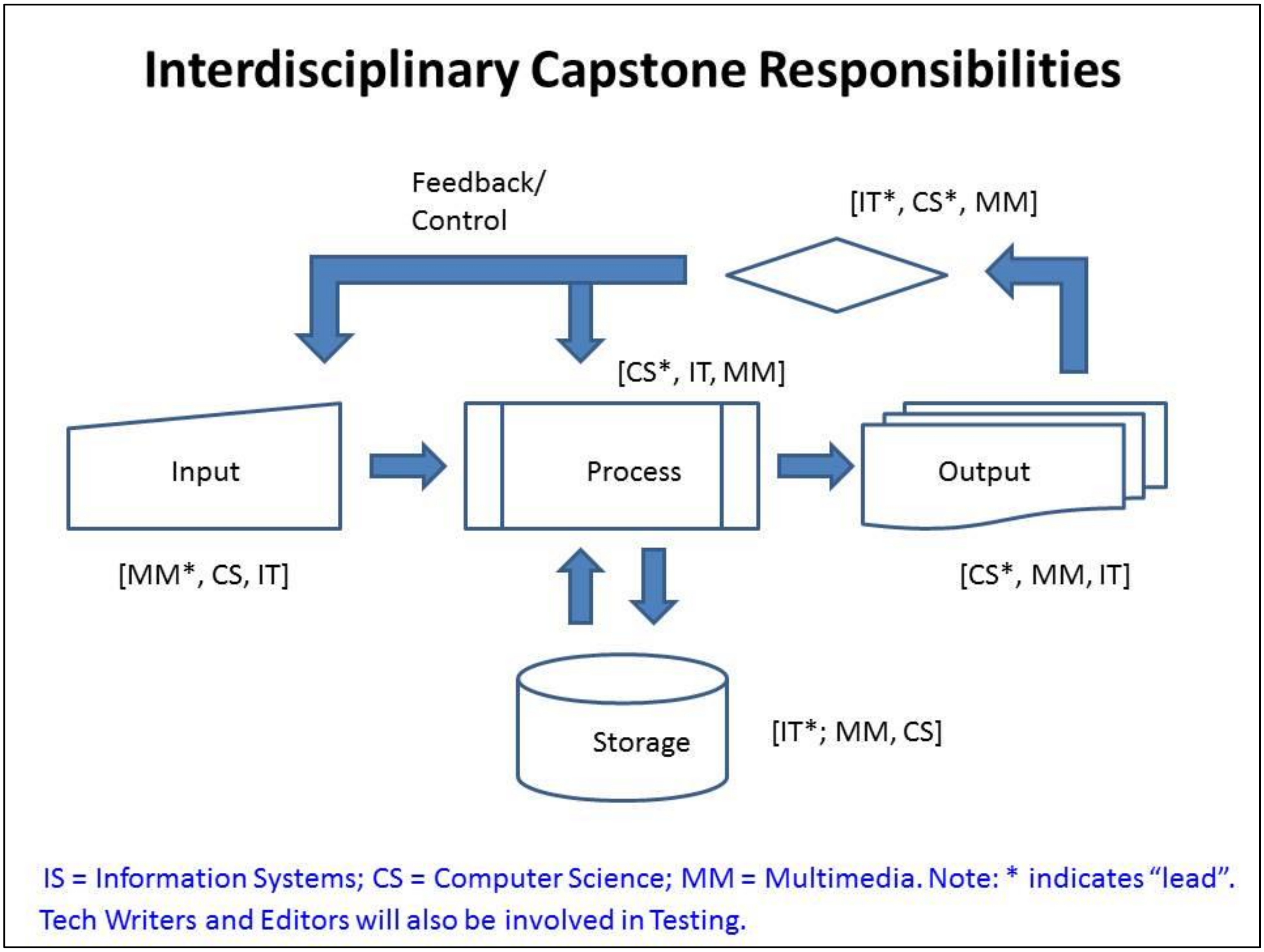

Interdisciplinary Capstone Responsibilities
Feedback/ Control
[IT*, CS*, MM]
[CS*, IT, MM]
Input
Process
Output
[MM*, CS, IT]
[CS*, MM, IT]
Storage
[IT*; MM, CS]
IS = Information Systems; CS = Computer Science; MM = Multimedia. Note: * indicates "lead".
Tech Writers and Editors will also be involved in Testing.

Figure 2. Web Project Components and Procedures

2.3.1 Requirements

After the teams were integrated, a brief synopsis of the project was provided to the students along with a milestone list. From this, students developed interview questions for the client. This interview took place in the third class period after team formation. Based on the results of the synopsis and interview, students obtained a signed Statement of Work (SOW) [3] and started their analysis portion of the project.

2.3.2 Analysis

Students utilized various modeling tools integrated within their CASE tool to determine the solution to the business problem provided by the client. These tools included Data Flow Diagrams, Entity Relationship Diagrams, and UML tools with Use Case diagrams. An example information flow diagram is shown in Figure 3 [3].

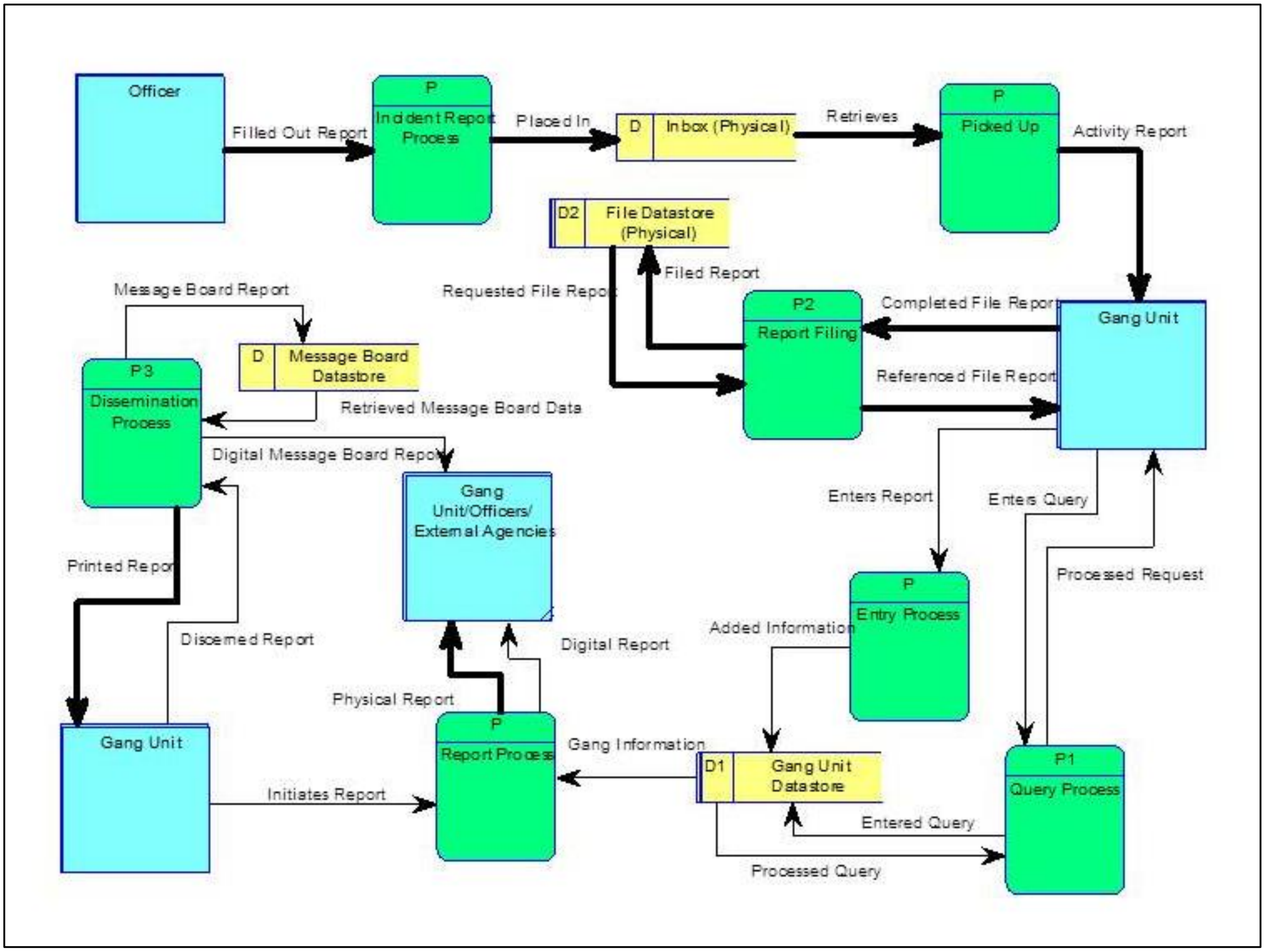


Figure 3. System Information Flow Diagram

2.3.3 Design

Instructors acted as "living" reference books, as they monitored and approved each stage of development. After completing Data Flow Diagrams and Conceptual Model Entity Relationship Diagrams, students gave an IPR to the client and the instructors. Other faculty and administration were invited to attend. Client representatives provided feedback and verified their requirements [4]. Necessary modifications were then made. An example flow chart for a proposed login page is shown in Figure 4.

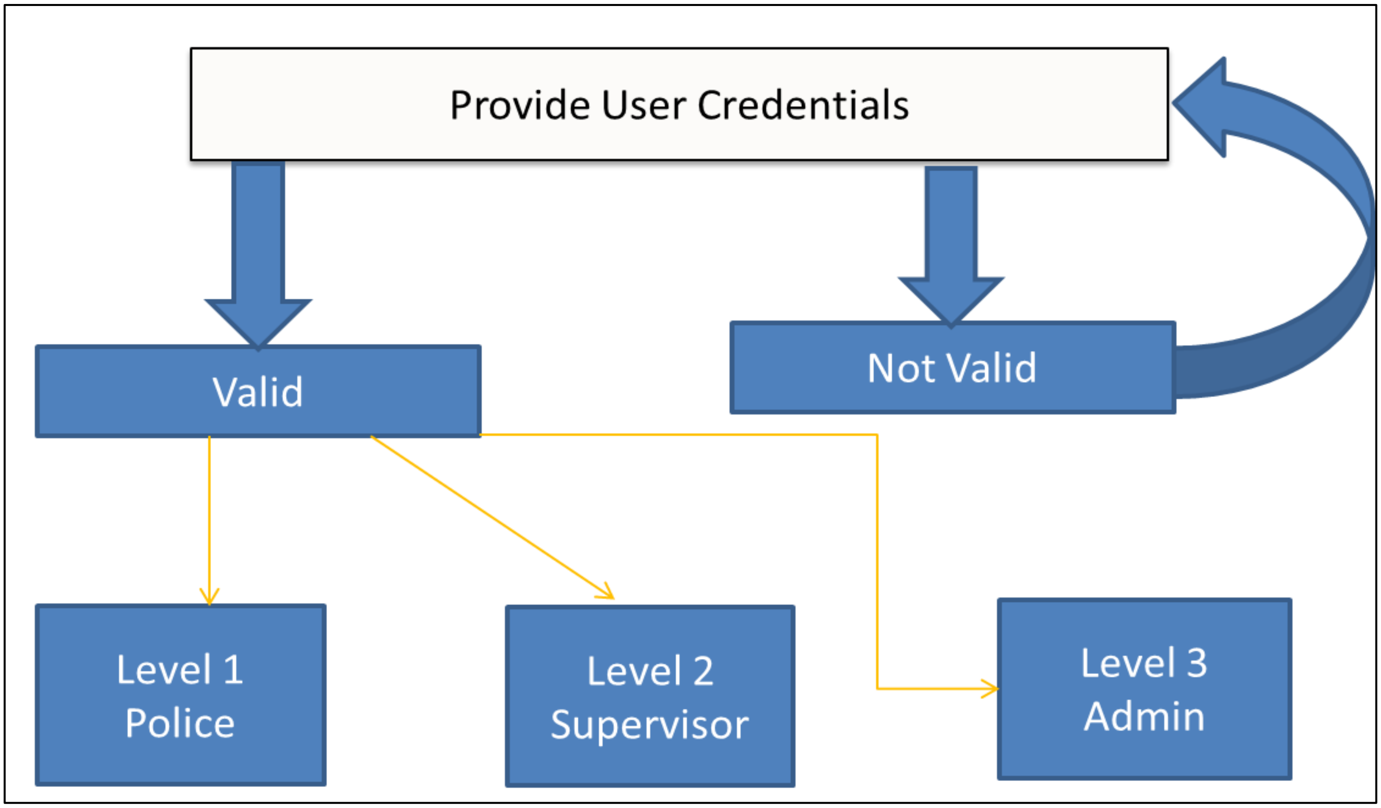


Figure 4. Login Page Design

2.3.4 Implementation

Implementation of the system took place in three phases. IT students completed database design. The CASE tool used provided the SQL DDL script, which was then used to create the backend. Once the backend was running, CS students coded web pages using HTML/CSS, PHP/JavaScript, and MySQL [4]. MM students used these pages as the basis for the frontend. At this point, a second IPR was then conducted, demonstrating a proof of concept to the client.

2.3.5 Testing

Student teams continually tested the system throughout the development of the project. This included both non-executable and executable testing. Non-executable testing consisted of general rehearsal, modeling how users would navigate through the system. Executable testing consisted of locating any logic or syntax errors in the coding process [2, 4].

2.3.6 Validation

The students' final exam was a live presentation of their product to the client. The client was able at that time to determine which team's project would be implemented on the client server. In the case of the Gang Unit database, one team's project was chosen. However, the client also chose some functionality from the other team's deliverable. The City of Lawton's IT staff were given copies of both, and the functionality from the one transferred to the other before full implementation. As a side note, when this occurs, it usually provides for an internship opportunity for one or more students, which was the case here.

3 Discussion

Running an interdisciplinary capstone course as a "virtual" software company is a winning scenario for the students, clients, university, and the professors involved. Here are just a few of the benefits reaped from this type of classroom environment.

- **Students obtain actual software development experience:** Students are provided with real-world software development experience and receive a letter of recommendation from the client for their efforts. Additionally, clients can look at potential employees and determine those that could fit into the client organization.
- **Students experience integrated teams:** Much of software development today is accomplished in teams. In many cases, these teams are composed of different disciplines. In this case, IT, CS, MM, and ENG students must work together to solve the client problem and produce a viable product.
- **Students develop better communication skills:** Soft skills development (oral and written) may not be a main

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

focus in many computing departments. However, research has shown that these skills are of critical importance to students entering the job market [5]. This course forces students to use both oral and written communication continuously throughout the development process. Students also are able to work with technical writers who enforce these skills.

- **Client organizations utilizing this course obtain products that have required functionality at a minimal cost:** By utilizing the skill sets of the students, organizations can obtain necessary software that may otherwise be unaffordable. As a specific instance of this, this product is in now in use by the Lawton Police Department, and has saved the city of Lawton an estimated $100,000 [4].
- **Faculty is forced to maintain their skill sets.** Faculty must be able to continually respond to student limitations in order to ensure project success.
- **This type of class provides the university with unique service/experiential learning opportunities for its students.** Service and experiential learning are becoming increasingly important to universities in southwest Oklahoma. This course also provides a mechanism for partnering with outside agencies within the regional area [1].

4 Conclusion

Real systems software development and implementation has now become the norm for this particular capstone course. This approach provides students with opportunities to grow in their skills and learn how to work in integrated teams to ensure project success. The approach also enhances networking between the university and local organizations, thereby continuing to provide significant positive social impact [4].

5 References

[1] ***The Cameron University Green Website Project-Part 1: Service Learning in the Fall 2009.*** Mike Estep, Kenneth David Smith, Chao Zhao, and Tom Russell, *International Journal of Education Research* , Volume 5 No. 2, Summer 2010.

[2] ***Object-Oriented and Classical Software Engineering***, 8th Edition, Stephen R. Schach, McGraw Hill, 2011.

[3] ***Information Systems Solutions A Project Management Approach,*** 1st Edition, Van Horn, Schwartzkoff, and Price, McGraw-Hill, 2006.

[4] ***Teaching Software Engineering Through a Real-World Project: A new Approach.*** Zhao, C., Estep, M., & Smith, K.D. WorldComp 2013, Proceedings of the 2013 International Conference on Software Engineering Research & Practice (SERP'13), Las Vegas, NV, July 22-25, 2013.

[5] ***Examining Quantity and Type of Communication in Contributing to IS Project success.*** Smith, K. and Merritt, K., Issues in Information Systems, Volume XI, No. 1, 2010.

A Teaching Methodology with Frame Buffer and Monitor Design in the Study of Hardware Track in Computer Science

Hassan Farhat
Computer Science
University of Nebraska at Omaha

Abstract- *While the study of computer design is covered extensively in the hardware track of computer science, minimal actual design of the remaining units required in image display is covered. Designs of simple CPUs with needed memory can be accomplished towards the end of a first course in digital design. Alternatively it can be covered during the early part of a second course on digital and computer architecture. Due to time constraints, the study of video controllers and frame buffers needed in the interface to a monitor is normally covered in advanced classes in computer engineering and is often skipped in computer science. Instead, computer simulation of designs or use of seven-segment-displays is often used to verify computer design correctness.*

In this paper we provide pedagogy for including frame buffer design topics in conjunction with CPU design. The proposed work integrated with previous work on computer design provides a practical sequence of topics to cover from design of a CPU to monitor display. To meet the time constraints imposed by the number of classes in hardware track, unlike discussion of design found using hardware descriptions languages, we provide a methodology of design at the schematic (gate) level. The proposed designs form a complete educational system of a computer with a frame buffer that can be covered in a two 3-credit hour course sequence.

The contribution of the paper is educational in nature. The paper provides a pedagogy of teaching a somewhat advanced design of topics during a first and a second course in hardware in computer science. We provide methodology of teaching designs of a simple buffer in a first course in the hardware track. These topics can then be used in conjunction with CPU designs in a second course on computer design.

(KEYORDS: Digital Design, Instruction Set, Video Controller, Frame Buffer, Computer Organization)

1 Introduction

The study of digital circuits from simple gates to central processing units and to complete computer design is found in computer science as well as computer and electrical engineering. To reduce the complexity of design and analysis, the digital design aspects are studied at several levels. It is covered at the lowest level, the switch and chip design level [1]. It is covered as well at the next higher level, the digital logic level [2] to [7]. At this level, gates are used as building blocks that are realized as several switches from the chip or switch level. Gates are then combined with registers and Arithmetic Logic Units (ALUs) and studied at the computer organization level [8, 9]. The computer architecture [10] to [13] level combines units from the organization level into processors, multi-level memories, and input/output interface units.

The study of hardware track in standard computer science curricula may include the design of a microprocessor. While this design can be accomplished towards the end of a first course or early part of a second course in digital and computer design, the designs of frame buffers and video controllers are normally skipped. The design of such units is normally covered in advanced undergraduate courses in computer and electrical engineering.

In earlier work we incorporated two design packages, Altera [14] and Multisim [15]. For both packages we looked at the design of a simple computer to include under curricula constraints. In [16] we showed the design with monitor interface using units provided by [18]. With the popularity of programmable logic devices and packages such as Multisim and Altera, we propose a teaching methodology that includes designs of sample frame buffers as part of the curricula. While these topics were outside the scope of teaching in a two 3-credit hour courses in digital design, we feel they can be included as part of a first course on digital design. To accomplish this, we show the design of both can be accomplished at the gate schematic level with no need for the overhead required in learning hardware description languages[1]. The frame buffer design is normally presented as a memory unit. Based on the desired interface with the simple computer design, we show the design can be built from components covered in a digital design course with two small ROMs (Read Only Memories). As a result, the design of the frame buffer and the video controller can be accomplished toward the end of a first course on digital

[1] A proposed video controller design at the gate level is found in [16].

design. Figure 1 shows the proposed contribution of this paper.

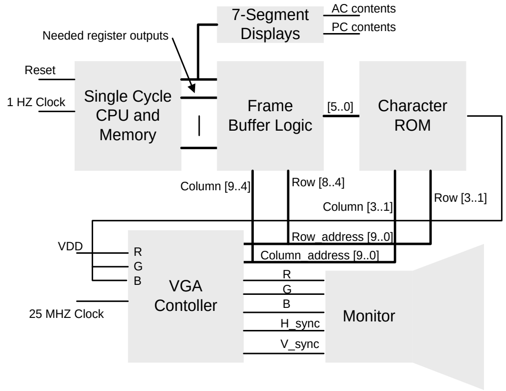


Figure 1: A schematic of the Complete Computer Design with Program Entered using Memory Initialization Files in Altera

The contribution is in proposing schematic realization of the frame buffer logic at the gate schematic level and providing a list of needed topics. Unlike previous work, we realize the design using simple schematic logic captures without the need to learn hardware description languages.

The paper is organized as follows. In section 2 we review the instruction set of the single-cycle computer. In section 3 we look at functionality of the frame buffer. Section 4 includes proposed designs and the experimental results. The conclusions are given in section 5.

2. Review of the Instruction Set of the Single-Cycle Computer

Designs of computers can vary depending on the size and type of the instruction set chosen. Popular texts include the design of a simple version of a MIPS Reduced Instruction Set Computer [8]. [13] is a classical text that discussed designs of accumulator based computers. We work with a similar simple design of an accumulator based computer. We also choose a single cycle design with a single addressing mode (direct addressing). Finally we choose a small set of instructions that conforms to instruction set completeness. A 4-bit opcode field is chosen for possible expansion. A small memory is chosen, 256 words. The rationale is to complete design process without having to keep track of all the needed details when compared to larger instruction sets. The complete instruction set of the computer, [16], is given in Table 1.

In [16] we gave simple examples of expanding the functionality of the computer by writing macros for additional logic and arithmetic operations. The instructions given above form a small subset from a popular earlier AC-based instruction set architecture (PDP 8).

Table 1: Simple AC based instruction set of computer design

Instruction	Meaning	Opcode
LDA XX	AC ← M[XX]	0XX
STA XX	M[XX] ← AC	1XX
ADD XX	AC ← AC+M[XX]	2XX
NAND XX	AC ← NOT (AC AND M[XX])	3XX
BUN XX	PC ← XX	4XX
SKZ XX	If AC = 0 then PC ← PC + 1	5XX
INC	AC ←AC + 1	6XX

Since the design is a single-cycle design, we need to complete instruction execution during the same cycle. Since the instruction needs to be fetched from memory, the instruction fetch and instruction execution must be completed in the same cycle. We accomplish this by having two memories, the instruction memory and the data memory. The presence of the data memory eliminates the need for a memory buffer register. A schematic of the simple design is shown in Figure 2, [16].

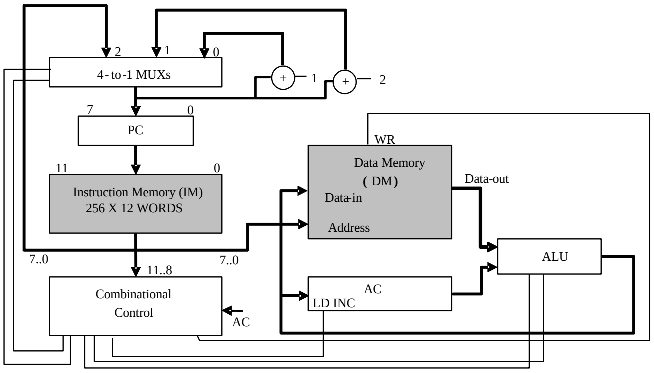


Figure 2: Schematic of the single-cycle computer

Table 2 shows how the control unit of the simple computer can be designed. Since the computer is a single cycle and since the instruction set is a small instruction set, the control unit can be realized as a small read only memory. The outputs of the ROM are the different control signals sent to the different units of the computer.

In the table, the PC_{MUX} is used to adjust the program counter (PC) value in case of branch instructions. From the instruction set, the branch unconditional (BUN X) and the increment and skip if zero (ISZ) instructions are the only two instructions that affect the value of the program counter. The multiplexer input to the program counter chooses from three options, the default increment PC and the previous two options.

3 The Frame Buffer and Needed Topics

We explore the design of the frame buffer interface as a

Table 2: Contents of ROM for simple computer design

	Control Address								Control Word Contents				
Instr.	AC_{11}	AC_{10}	……	AC_0	IM_{11}	IM_{10}	IM_9	IM_8	PC_{MUX}	ALU	AC_L	AC_{INC}	WR
LDA X	x	x	……	x	0	0	0	0	00	00	1	0	0
STA X	x	x	……	x	0	0	0	1	00	01	0	0	1
ADD X	x	x	……	x	0	0	1	0	00	10	1	0	0
NAND X	x	x	……	x	0	0	1	1	00	11	1	0	0
BUN X	x	x	……	x	0	1	0	0	10	xx	0	0	0
SKZ	0	0	……	0	0	1	0	1	01	xx	0	0	0
SKZ	1	x	……	x	0	1	0	1	00	xx	0	0	0
SKZ	x	1	……	x	0	1	0	1	00	xx	0	0	0
……	……	……	1	……	0	1	0	1	00	xx	0	0	0
SKZ	x	x	……	1	0	1	0	1	00	xx	0	0	0
INC	x	x	……	x	0	1	1	0	00	xx	0	1	0
Not used	x	x	……	x	x	1	1	1	xx	xx	x	x	x

small circuit realized at the schematic level from units covered in a digital design course. The intent is to: a) introduce the students to the concepts of frame buffers early in the hardware curricula, and b) explore the concepts of significant saving in hardware based on the application.

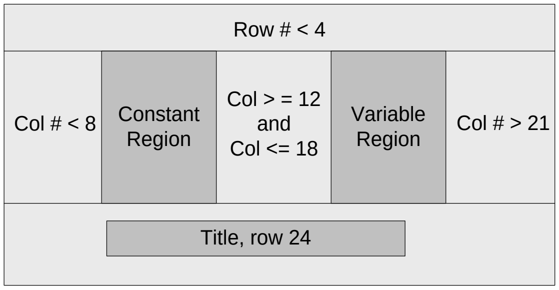


Figure 3: VGA display divided into four regions, background region colored as ░

The VGA monitor for our application is used to display the current instruction and the computer register contents. We refer to Figure 3 for illustration. To design the frame buffer we divide the monitor area into four regions: 1) the background region, 2) the constant region, 3) the title region, and 4) the variable display region. The background region display the background color. The title region is used to display the title of the design. The constant region displays 4 character words representing registers, arithmetic logic unit, the current instruction and data memory labels. The variable region displays the contents of the corresponding registers, the ALU, and instruction and data memory. The variable values are the only part of the display that changes.

In a traditional design, the frame buffer is composed of a memory unit with contents that mirror the contents displayed on the screen. Here, however, based on the application, we choose a combination of a small memory and standard logic gates to realize the frame buffer design. The frame buffer produces the needed pixel values for a 640×480 standard VGA monitor. The character sizes displayed are 16 by 16 pixels. As a result the display is composed of 30 rows (480/16); each row is 40 characters. By incorporating a small character ROM using concepts of lookup-tables from early designs, one can reduce the memory requirements.

We reduce the memory requirements further by: a) using two lookup tables, and b) determining the needed logic gates design for region determination. A partial schematic of the frame buffer design is shown in Figure 4. The design is done in the Altera MAX+PLUS II package.

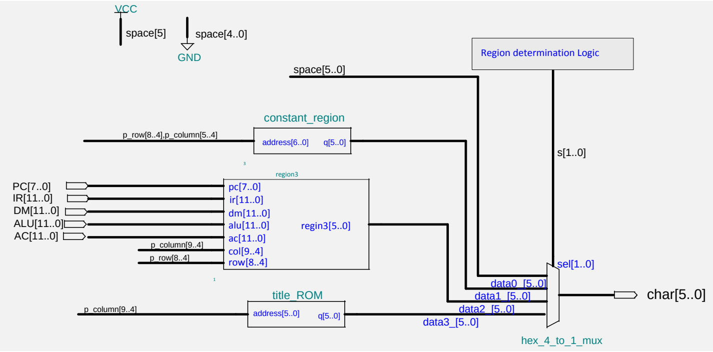
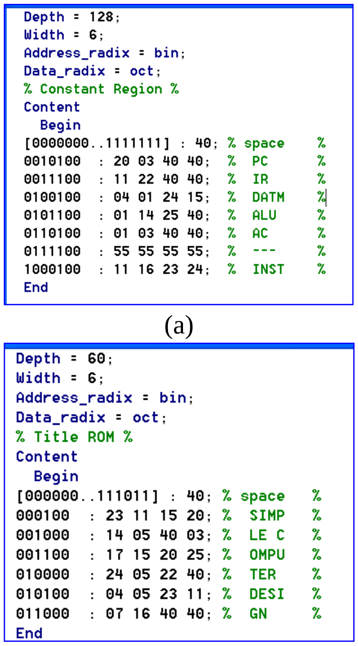


Figure 4: Partial Schematic of the Frame Buffer Design

As can be seen the traditional memory realization of a frame buffer is replaced with gate schematic and two small ROMs (constant region includes a ROM as well). Only region 3 receives inputs from the simple computer design. The hex_4_to_1_mux selects which of 4 data busses to route to the output. The select lines values are computed by the region determination logic block diagram. The logic determination block uses a set of comparators with column and row values as inputs obtained from a standard VGA controller. The title ROM and the constant region ROMs are small memory units used to display the proper characters within each of the regions. The space[5..0] data input displays the space character (ASCII 40_8). The memory initialization files for the constant and title regions are shown in Figure 5 (a) and 5 (b) respectively.



```
Depth = 128;
Width = 6;
Address_radix = bin;
Data_radix = oct;
% Constant Region %
Content
  Begin
[0000000..1111111] : 40; % space    %
0010100  : 20 03 40 40;  %  PC      %
0011100  : 11 22 40 40;  %  IR      %
0100100  : 04 01 24 15;  %  DATM    %
0101100  : 01 14 25 40;  %  ALU     %
0110100  : 01 03 40 40;  %  AC      %
0111100  : 55 55 55 55;  %  ---     %
1000100  : 11 16 23 24;  %  INST    %
End
```

(a)

```
Depth = 60;
Width = 6;
Address_radix = bin;
Data_radix = oct;
% Title ROM %
Content
  Begin
[000000..111011] : 40; % space   %
000100  : 23 11 15 20; %  SIMP   %
001000  : 14 05 40 03; %  LE C   %
001100  : 17 15 20 25; %  OMPU   %
010000  : 24 05 22 40; %  TER    %
010100  : 04 05 23 11; %  DESI   %
011000  : 07 16 40 40; %  GN     %
End
```

(b)

Figure 5: Memory Initialization Files using Altera MAX+PLUS

We illustrate the design of region 3 and show the design can be accomplished from standard topics covered in digital design. Two sets of multiplexers are used; one set is used to determine the proper row of the display and another set is used to display the proper column. Figure 6 show a partial schematic of the column multiplexers. As can be seen, the bulk of the realization is done using multiplexers.

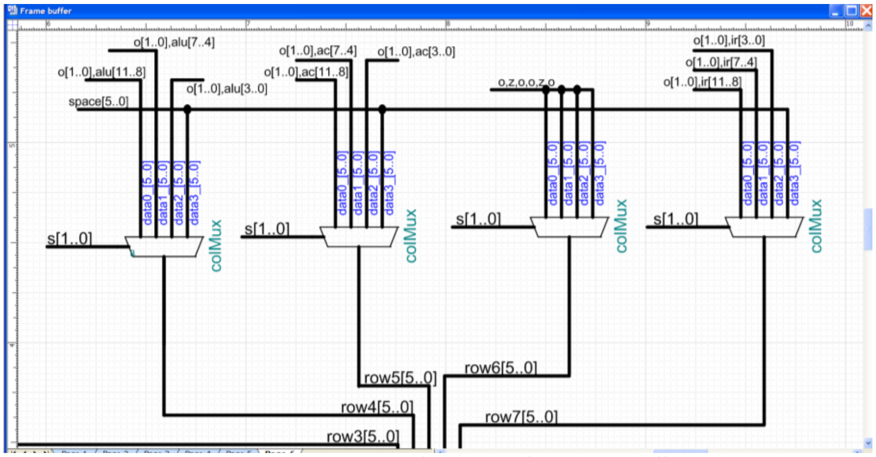


Figure 6: Partial Design of Region 3 of the Frame Buffer

In the above schematic, the S[1..0] selects from among 4 columns to display on the. The outputs of the multiplexers are used as data inputs to a row select multiplexer. The row multiplexer then selects one of the character codes available at its inputs. If the region determination logic selects the variable region this code is forwarded to the character ROM unit.

Due to space limitation, we skip further details of the design. Instead, we look at a sample case in the design where the AC contents are to be displayed. By referring to Figure 6, we note that the s[1..0] select lines are such that 00 selects the leftmost column and 11 selects the rightmost column. Hence the AC contents are placed on the multiplexer data inputs from most significant (multiplexer data 0 = ac[11..8]) to least significant (multiplexer data 3 = ac[3..0]). As can be seen from the figure, actual inputs have o[1..0] = 11 appended from the left. This is used so as to display the variable data properly [18, p. 150]. In [18] the character ROM table is such that the hexadecimal character codes 0 through F have the octal code 60 through 77. As a result, to display the proper hexadecimal value of a register we append 11 to the left part of the code. For example, an AC[7..4] = 1010 is displayed as hexadecimal A by appending 11 to the left part of the binary value 111010 is octal 72 (character A).

4 Experimental Results

The computer design and monitor interface was completed in MAX+PLUS II. To experiment with the design, the complete design was downloaded to the UP2 board supplied by Altera as part of university educational program. The board contains two programmable logic devices the MAX^R 7000 Complex Programmable Logic Device (CPLD) and a FLEX 10K Field Programmable Gate Array Device (FPGA). The computer design was downloaded to the FPGA part of the UP2 board. This part contains two seven-segment displays and a VGA interface used to control a monitor display when connected to the interface.

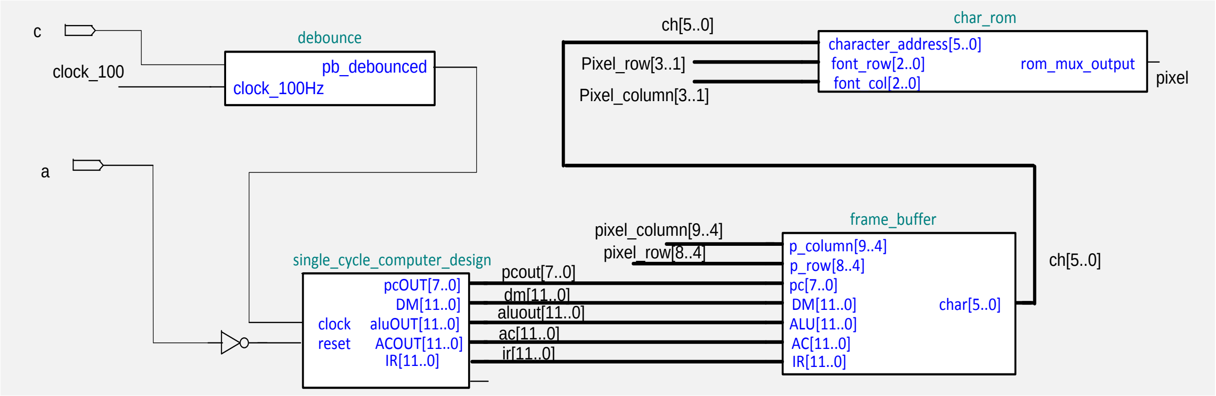


Figure 7: Partial design of computer and frame buffer interface

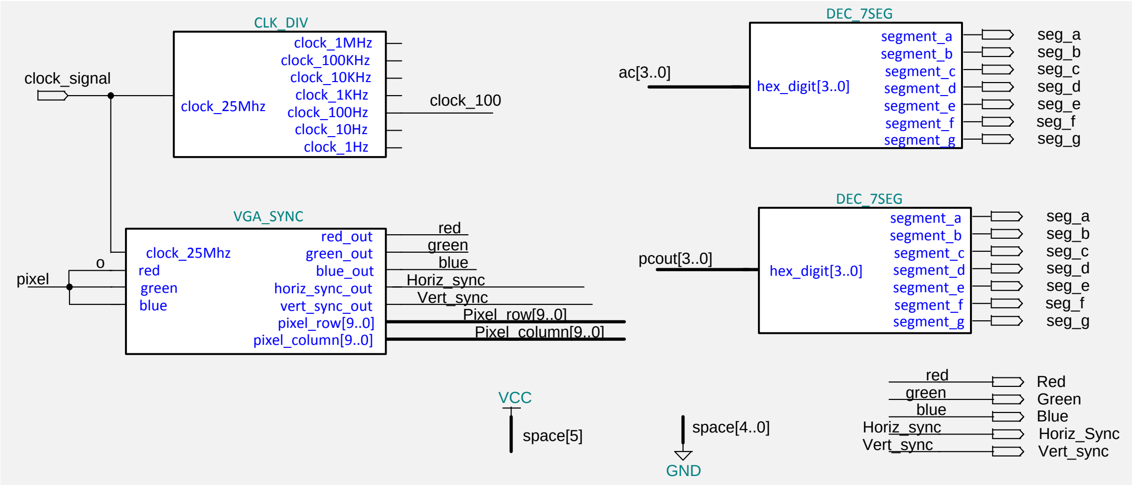
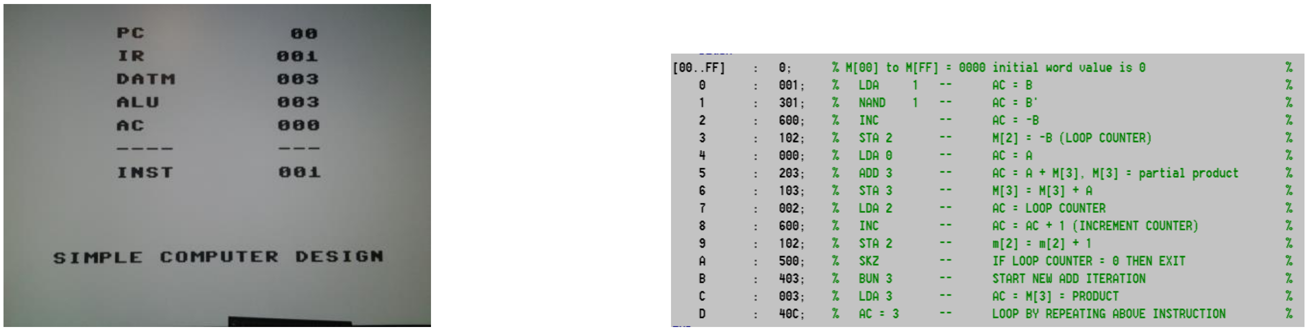


Figure 8: Remaining design of computer and frame buffer interface

Figure 7 and Figure 8 show the high-level block design in MAX+PLUS II.

The design of the auxiliary units is found in [17]. We have implemented the design and downloaded the design to the Altera UP2 board. A partial screen capture of the monitor display is shown in Figure 9(a). The display shows the register contents of a multiplication program shown in Figure 9(b). Pushbuttons on the up2 board are used to clock the design or to reset the program counter. The program counter is reset to the first instruction of the multiplication program. On clocking the design we step through the instruction set of the sample program. The accumulated result is stored in data memory location 3.

In the screen capture shown in Figure 9, column 1



```
PC      00
IR      001
DATM    003
ALU     003
AC      000
----    ---
INST    001


SIMPLE COMPUTER DESIGN
```

```
[00..FF]  :  0;     % M[00] to M[FF] = 0000 initial word value is 0          %
   0      :  001;   %  LDA    1  --   AC = B                                 %
   1      :  301;   %  NAND   1  --   AC = B'                                %
   2      :  600;   %  INC       --   AC = -B                                %
   3      :  102;   %  STA 2     --   M[2] = -B (LOOP COUNTER)               %
   4      :  000;   %  LDA 0     --   AC = A                                 %
   5      :  203;   %  ADD 3     --   AC = A + M[3], M[3] = partial product  %
   6      :  103;   %  STA 3     --   M[3] = M[3] + A                        %
   7      :  002;   %  LDA 2     --   AC = LOOP COUNTER                      %
   8      :  600;   %  INC       --   AC = AC + 1 (INCREMENT COUNTER)        %
   9      :  102;   %  STA 2     --   m[2] = m[2] + 1                        %
   A      :  500;   %  SKZ       --   IF LOOP COUNTER = 0 THEN EXIT          %
   B      :  403;   %  BUN 3     --   START NEW ADD ITERATION                %
   C      :  003;   %  LDA 3     --   AC = M[3] = PRODUCT                    %
   D      :  40C;   %  AC = 3    --   LOOP BY REPEATING ABOVE INSTRUCTION    %
```

Figure 9: Screen Capture of Monitor in Standard VGA

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

contains the constant region. This region contains: the computer registers, DATM where the result of multiplication is stored in data memory, the three dashes, and the current instruction. The title region displays the simple computer design title. The variable region is column 2, where the contents of the program counter, instruction register, arithmetic logic unit, accumulator, current instruction and result in data memory is displayed.

5 Conclusion

While the study of computer design is normally covered in a first or second course in the hardware track of computer science, minimal or no coverage of peripheral units is studied. Topics such as actual VGA or frame buffer designs are normally reserved for upper classes in computer engineering. In this paper we looked at how the study of such units can be incorporated early in digital design classes based on the knowledge needed to accomplish the design. In particular we looked at the design of a frame buffer at the schematic level without the need to employ hardware description languages.

References

[1] N. Weste and D. Harris. “CMOS VLSI Design: A Circuits and Systems Perspective 4th edition”. Addison Wesley, 2011.

[2] S. Brown and Z. Vranesic. “Fundamentals of Digital Logic with VHDL Design with CD-ROM, 3rd edition”. McGraw-Hill, 2009.

[3] T. Floyd. “Digital Fundamentals: A System Approach”. Prentice Hall, 2013.

[4] W. Kleitz. “Digital Electronics A Practical Approach with VHDL, 9th edition”. Prentice Hall, 2012.

[5] M. Mano and M. Cilette. “Digital Design, 5th edition”. Prentice Hall, 2013.

[6] R. Tocci, N. Widmer and G. Moss. “Digital Systems Principles and Applications, 11th edition”. Prentice Hall, 2011.

[7] J. Wakerly. “Digital Design Principles and Practices 4th edition”. Prentice Hall, 2006.

[8] D. Patterson and J. Hennessy. “Computer Organization and Design, Revised 4th edition”. Morgan Kaufmann, 2011.

[9] W. Stallings. “Computer Organization and Architecture, 9th Edition”. Prentice Hall, 2013.

[10] D. Harris and S. Harris. “Digital Design and computer Architecture, 2nd edition”. Morgan Kaufman, 2013.

[11] J. Hayes. “Computer Architecture and Organization 3rd edition”. McGraw-Hill, 1998.

[12] J. Hennessy and D. Patterson. “Computer Architecture: A Quantitative Approach 5th Edition”. Morgan Kaufmann, 2011.

[13] M. Mano. “Computer System Architecture, 3rd Edition”. Prentice Hall, 1993.

[14] Altera, www.Altera.com.

[15] Electronics Workbench, www.electronicsworkbench.com.

[16] H. Farhat. “A Simple Computer Design with Monitor Interface: Integrating Hardware and Software in Early Research in Computer Science”. Proc. ICSNC, 218-222, 2008.

[17] H. Farhat. “Including Computer Design in Early Hardware Study of Computer Science”. Int'l Conf. Frontiers in Education: CS and CE, FECS’13, pp. 263-267, July 2013.

[18] J. Hamblen and M. Furman. “Rapid Prototyping of Digital Systems, a Tutorial Approach”. Kluwer Academic Publishing, 2001.

Capstone Experience with Watermarking

Suhair Amer
Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO, USA

Abstract – *This paper documents a capstone experience of one group. The Students were required to investigate watermarking and they were able to enhance the performance of the investigated system. The paper maps student activities to the university's learning outcomes for undergraduate students. In addition, it lists how their product was assessed.*

Keywords: capstone, watermarking, senior project.

1 Introduction

1.1 Capstone Experiences Defined

The American Association of Colleges and Universities define the senior capstones as requiring students who are "nearing the end of their college years to create a project of some sort that integrates and applies what they've learned. The project might be a research paper, a performance, a portfolio of "best work," or an exhibit of artwork. Capstones are offered both in departmental programs and, increasingly, in general education as well." [Value 2011] [High Impact 2011]

Berheide (2007) explained that capstone experiences are an effective way to assess student learning. For example, capstone products can be assessed by:

- Encouraging students to present their projects at conferences.
- Encouraging students to submit their papers to contests.
- Inviting external evaluators, such as industry experts, potential employers, etc. to grade projects.
- Requiring students to publicly present their work.
- Evaluating senior seminar papers based on program quality and learning outcomes.

1.2 University Learning Outcomes for Undergraduate Students

Next is a list of several university outcomes for undergraduate students from which an institution can adapt depending on their desired capstone experience [Campbell and Lambright 2011] [Lockhart and Borland 2001][Todd, R., & Magleby, S. (2005)]:

1. "Master the depth of knowledge required for a degree, including the ability to:
   a. Articulate disciplinary and interdisciplinary theories, concepts, principles, skills and practices.
   b. Synthesize knowledge across courses and other experiences.
   c. Apply knowledge from core curriculum courses, discipline-based courses, and other experiences in a range of contexts to solve problems and make decisions.

2. Demonstrate critical thinking, including the ability to:
   a. Evaluate, analyze, and integrate information from a variety of sources.
   b. Use appropriate strategies and tools to represent, analyze and integrate information.
   c. Develop critical, reasoned positions.

3. Communicate effectively, including the ability to:
    a. Demonstrate effective oral communication skills (which could include the use of languages such as American Sign Language for those who do not communicate orally).
    b. Demonstrate effective writing skills.
    c. Demonstrate effective nonverbal communication skills (which could include appropriate use of performance, design or representations such as maps, tables and graphs).
    d. Listen actively and critically.
    e. Effectively communicate original and creative ideas.

4. Practice personal and social responsibility, including the ability to:
    a. Practice ethical leadership.
    b. Recognize an ethical dilemma and apply rational decision-making in order to address it.
    c. Choose ethical courses of action in research and practice.
    d. Acknowledge and address the consequences of one’s own actions.
    e. Engage in local and global civic activities.

5. Demonstrate social, cultural, and global competence, including the ability to:
    a. Live and work effectively in a diverse and global society.
    b. Articulate the value of a diverse and global perspective.
    c. Recognize diverse economic, political, cultural and religious opinions and practices.

6. Prepare to engage in lifelong learning, including the ability to:
    a. Exhibit the skills necessary to acquire, organize, reorganize and interpret new knowledge.
    b. Show proficiency in current technologies and the ability to adapt to emerging technologies.
    c. Recognize and participate in activities that enhance wellness of body, mind and spirit.
    d. Formulate a plan of personal goals for continued professional growth.
    e. Demonstrate intellectual curiosity.

7. Work collaboratively, including the ability to:
    a. Participate effectively in teams.
    b. Consider different points of view.
    c. Work with others to support a shared purpose or goal.”

2 Capstone Project details

2.1 Project details

The students were required to investigate and implement an application to authenticate/ secure bio-metric images using watermarking based on the algorithm in [Thampi and Jacob 2011]. In general, watermarking is used to increase security of biometric data. For example, in the reversible watermarking, we can restore the original image and the retrieve the watermark. Reversible watermarking maintains high quality of biometric data. In addition, the Rational Replacement of Least Significant Bit (LSB) is a reversible watermarking scheme for biometric images. Peek-Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) index are methods used for quality measurement of biometric data.

2.2 Students' Findings and proposed improvements

The students in this group managed to polish the algorithm in [Thampi and Jacob 2011] into something they felt will work better.

They proposed a two-stage watermarking and encryption system that will be capable of producing a final image that should be so unobtrusive that it would be visibly indistinguishable from the original to the untrained eye by only working on the least significant bit. This system have two stages:

Stage one: Data embedment – An optional step that can be used to inject arbitrary data through various methods elaborated in the document that would benefit various scenarios.

Stage two: Image signing – This step utilized public key cryptography to ensure authenticity. The image is hashed and encrypted with the issuer's private key. This enables recipients to verify the image source and contents.

This system is, overall, more secure than the algorithm in [Thampi and Jacob 2011] and only exhibits a weakness that is inherent to the unobtrusive security scheme as a whole, so any system created will be susceptible to this specific attack.

The students agreed that this system would be much better even without the attack correction, and should not create a deadline issue. The system would use a combination of XOR and public key cryptography to create a system capable of embedding watermark information into an image and providing a method by which to verify the issuer and integrity of the image.

Next is a much more detailed explanation.

3 Enhanced system

This section details the students' proposed enhanced solution. Additional details can be found in [Bates et al. 2013].

3.1 Terms

Issuer – The person or company that is issuing the secured image

Recipient – The third party that is using the secured image, this may also include an attacker that has access to the systems and data of the third party

LSB – Least significant bit

LSBS – Least significant bit stream – The stream of bits created by all of the LSBs in the image – This is the only data that is changed, preserving overall visual quality

Image i – The original image

Image j – The image created from stage one that contains embedded data

Image k – The secured image created from stage two. This image is distributed to the recipient for usage

Method H – Hashing algorithm used to create hash h

Hash h – The hash of image j used to verify authenticity

Hash l – The hash generated by the recipient from their decrypted version of k that was given to them

Keys e and f – The issuer's private and public keys, respectively

Key d – Encryption method to an additional cryptosystem, possibly a public key system.

Message m – The data embedded into i to create j

3.2 Stage One: Data Embedment

In this stage, the original image can have its data inserted into the LSBS through means that can either be lossy or lossless. Embedding data into the LSBS will prevent the recipients from being able to recreate the original image. This stage is optional and the original image can be passed directly to stage two if desired. This stage includes two steps.

In step one, Lossless Embedment using XOR was performed. Message m is embedded into the LSBS by replacing the image LSBS with an LSBS created by XORing m with the original image's LSBS. This method will embed data that will be invisible to the recipients. Message m must remain a secret to the recipient, as it can be used to recreate the original image.

Method: $LSBS_j = LSBS_i \oplus m$

Example: The issuer can imbed a fingerprint that is unique for each recipient. This may also be used to discover the source of an information leak.

In step two, Lossy Embedment was performed. This will generally prevent even the issuer from restoring the original image, but allows for much more flexible message options. This is achieved by performing two types of insertions: Plaintext insertion and cyphertext insertion.

In the plaintext insertion, the LSBS of the original image is overwritten with a plaintext message m. This message will be visible to anyone

Method: $LSBS_j = m$

Example: The issuer embeds additional information such as a timestamp into the original image.

In the cyphertext insertion, the LSBS of the original image is overwritten with a version of message m that had been encrypted, ensuring that m is only accessible to those with the decryption method.

Method: $LSBS_j = d(m)$

Example: The issuer embeds sensitive information such as a name or ID number into the corresponding person's photo.

3.3 Stage Two: Image Signing

In this stage, the issuer both signs and generates verification data from image j. This will include: hashing and signing.

In the hashing stage, the image is run through a hashing algorithm to create hash h which will be delivered to the recipient along with image k.

Method: $H(j)$

In the signing stage, the LSBS of image j is replaced with a version of the LSBS that have been signed with the issuer's private key to create image k.

Method: $LSBS_k = e(LSBS_j)$

3.4 Verification

To verify the authenticity, the recipient only needs to decrypt, rehash, and compare the hashes.

In the decryption stage, the received image k is decrypted using the issuer's public key to recover image j.

Method: $j = f(k)$

In the re-hashing stage, the recovered copy of image j is hashed using the same hashing algorithm of the issuer to create hash l.

Method: $H(j)$

In the comparing stage, the two hashes are compared, if they match, the image has not been tampered with.

Method: $h = l$

3.5 Usage

Due to the way the image is signed, the recipient will be able to conduct image comparisons with signed image k as well as with decrypted image j, although to access any potential embedded information, the recipient must first decrypt k.

3.6 Weakness

This system, as well as any system that is only partially encrypting a message, is susceptible to a man in the middle attack where the attacker replaces any data that was not signed with their own. In this specific system, the signing system can be defeated by first decrypting k with the issuer's public key to get j. Once the attacker has j, they can replace any data not signed in k (all but the LSBS) and hash their new image with the hash method used by the issuer. The attacker then replaces the LSBS of j with the LSBS of k, the only signed part of the image. Now the attacker has a modified signed image as well as a correct hash that they can do with as they please.

3.7 Solution

This attack can be prevented by signing the whole image, but the encrypted image k would no longer be able to be used for image comparison without first being decrypted into image j.

4 Student Reflection

All students agreed that they learned "a lot" from the project. First, they learned about biometric encryption and algorithms. Second, and most importantly, "the project helped" them "develop" their "skills of working with a team and completing project requirements in time." In other words they learned the value of team work and personal commitment. They also stated that "it became quickly apparent that we all can have very different backgrounds in software development, and we need to acknowledge and work with those differences to accomplish anything meaningful." [Bates et al. 2013]

The students also indicated that the use of tools like Drop boxes and Google Drive helped them share any new resources instantly and having the remaining group members review their mistakes. [Bates et al. 2013]

In this project, some students had to learn a new programming environment within a short time frame and to get through the frustration of understanding why something did not work. Some stated that looking at the problem from different angles helped them solve their problems. [Bates et al. 2013]

They also indicated that the software development processes taught as part of the Software engineering process helped them with their work flow. [Bates et al. 2013]

A couple of students stated that they had two more people than they needed for implementing the code. However, they did contribute in analysis, design, documentation and presentation but did not contribute much in implementation. [Bates et al. 2013]

Finally, the student who realized the shortcomings of the original algorithm, states that he is grateful for taking his math cryptography class. This is because he was able to analyze the security of the system and to design a better secure system. [Bates et al. 2013]

5 Conclusion

In this capstone experience the students were required to investigate a watermarking technique. They were informed about their project late December but they started the different stages of their project at the beginning of spring semester which is in late January.

Before implementation they were required to go through analysis and design phases and produce documents that had to be approved by client and course coordinator. The students worked in groups and presented their work in stages (analysis, design, and implementation) to the remaining groups. Utilizing their knowledge acquired from classes, internships, personal, etc. they implemented the watermarking algorithm. The students even went beyond what was required and came up with an enhanced solution to the algorithm which has been mentioned in this paper.

From the students' point of view, they enjoyed their experience. A couple of students did indicate that they were not engaged in programming as they would have liked to. However, in any agency, computer science graduates will not only be required to do programming but analysis and design are also important phases in development.

From the department's point of view, the group satisfied all Learning Outcomes for Undergraduate Students mentioned in section 1.2.

6 References

[Bates et al. 2013] Zach Bates, Jared Darr, Brandon Guttersohn, Corey Holthaus, and Will Starms. Capstone Experience Project: Invisible And Reversible Digital Image Watermarking. Southeast Missouri State University. 2013.

[Berheide 2007] Berheide, C. W. (2007). Doing less work, collecting better data: Using capstone courses to assess learning. Retrieved from http://www.aacu.org/peerreview/pr-sp07/pr-sp07_index.cfm

[Campbell and Lambright 2011]Campbell, D., & Lambright, K. (2011). How valuable are capstone projects for community organizations? lessons from a program assessment. Journal of Public Affairs Education, 17(1), 61-87.

[High Impact 2011] American Association of Colleges and Universities. (2011). High Impact Educational Practices. Retrieved from http://www.aacu.org/leap/documents/hip_tables.pdf

[Lockhart and Borland 2001]Lockhart, M., & Borland, K. (2001). Critical thinking goals, outcomes, and pedagogy in senior capstone courses. The Journal of Staff, Program Organization Development, 18(1), 19-26.

[Thampi and Jacob 2011]Sabu M. Thampi & Ann Jisma Jacob. Securing Biometric Images using Reversible Watermarking. Journal: International Journal of Image Processing. 2011. Vol: 5 Issue: 4 http://arxiv.org/ftp/arxiv/papers/1111/1111.1093.pdf

[Todd, R., & Magleby, S. (2005).]Todd, R., & Magleby, S. (2005). Elements of a successful capstone course considering the needs of stakeholders. European Journal of Engineering Education, 30(2), 203-214.

[Value 2011] American Association of Colleges and Universities. (2011). VALUE: Valid Assessment of Learning in Undergraduate Education. Retrieved from http://www.aacu.org/value/rubrics/index_p.cfm?CFID=31504336&CFTOKEN=44983028

Stimulating Software Security Education at Community Colleges Through Training Workshops

Akbar Siami Namin
Computer Science Department
Texas Tech University
Lubbock, TX, USA
akbar.namin@ttu.edu

Fethi A. Inan
College of Education
Texas Tech University
Lubbock, TX, USA
fethi.inan@ttu.edu

Rattikorn Hewett
Computer Science Department
Texas Tech University
Lubbock, TX, USA
rattikorn.hewett@ttu.edu

***Abstract*—We report our experience with training instructors and faculty scholars from two and four-year colleges in the software and information security related areas. The purpose of offering the workshop was to provide training for faculty and instructors who had less opportunity to learn about the state-of-the-art in software and information security research and education. The one-week long training in software security exposed the faculty and instructors to various aspects of cyber security research and practices. The workshop organizers have collected a rich set of data such as training sessions' logs, questionnaires, interviews, and knowledge acquired for the purpose of conducting quantitative and qualitative analyses. We report the structure and theme of the training sessions on software and information security, delivery methodology, and feedbacks received from the faculty and instructors, who participated in the training sessions.**

***Keywords*—Cybersecurity; faculty development, training workshop**

I. Introduction

It is reported that the United States needs to produce far more IT graduates and in particular students with Information Assurance (IA) capabilities [1]. Hoffman et al. [2] note that maintaining and protecting software systems and sensitive data require a proper and systematic education in information warfare. Not only educational institutions but also corporations must adopt the best practices and integrate cybersecurity practices and exercises into their curriculum, organizations and workflows. A survey published by the Information Security Magazine reports that U.S. colleges and Universities are ranked among the poorest protected against cyber-attacks.

Information assurance related areas such as operating system, information and data, network and Internet security are amongst the most important topics in security education [3]. In addition to these areas, we need also to include application and software security as an integral part of these predominant and important concepts. The needs for cyber security professionals in all aspects of education and business as well as homeland security and warfare demonstrate a clear demand for effective development of education and training programs with its grand intention to produce high quality and well-prepared workforce capable of protecting critical infrastructure.

Raising the awareness against cyber crimes and educating prevention techniques against potential cyber-attacks are important to the national defense and homeland security. The U.S. economic infrastructure, such as power and water plants, heavily depend on computing systems such as SCADA (Supervisor Control And Data Acquisition). The SCADA-based smart grids are vulnerable to many threats and are the potential targets for attackers. The importance of securing computer networks motivates this project with its target in preparing the future workforce needed for protecting the U.S. critical infrastructure against malicious eavesdroppers. For instance, network intrusion attempts occur approximately 250,000 per hour, i.e. 6 million times per day, according to the U.S. Government and Department of Defense [4]. In particular, financial transactions and payment card data are highly vulnerable to cyber-attacks [5]. According to CWE/SANS and OWASP (Open Web Application Security Project), SQL (i.e., top 10 web application vulnerability) and OS code injections, i.e. insertion of a SQL query or OS command via the input data from the client to the application, are among the top 25 most dangerous software errors that can lead to serious vulnerabilities in software. The objectives of training the software related workforce is to build more secure software systems and thus prevent adversaries from obtaining or tampering critical data.

The preparation of future cyber security professionals and workforce primarily depends on the availability of cybersecurity instructional courses as well as certifications and degree programs offered by higher education institutions. However, insufficient professional development specifically designed for faculty and instructors, who can create, develop, and integrate recent advancements of cybersecurity curriculum with hands-on experience, have been an escalating concern. Along with technological advances, cybersecurity is experiencing an explosion in growth with new and more complex threats, techniques, and applications. In order to achieve effective cybersecurity programs, instructors should be exposed to the recent advancement and the latest technology, skills, and confidence to effectively design and implement cybersecurity instructional lessons and course modules. Instructors, who are well prepared to cybersecurity education, are more likely to engage in security related teaching practices as compared to those who are less exposed to these kinds of activities.

We report our experience with designing, recruiting, and implementing training workshops on software and information security for the faculty scholars and instructors, who teach at two and four-year colleges across the southwest region of the United States. More specifically, we report our observations regarding the satisfaction level that the workshop participants reported. The paper provides an extensive investigation and assessment of the overall effectiveness of software security training program on participating faculty's learning, knowledge, satisfaction, and confidence. Throughout this paper, we refer to the faculty scholars and instructors who participated in our workshop as *participants*.

In addition to stimulating and promoting cyber security research and education through training workshops, we are also interested in learning about the impact, effectiveness, and the knowledge the participants gained through traditional training workshops. In other words, we are interested in understanding about the learning outcomes and impact of the workshop and whether the traditional training through workshops can address the nationwide needs in security experts.

There were several questions that motivated us to pursue the idea of offering training workshops: What were the learning outcomes for participating faculty after attending the training workshop on software and information security? Did participants gain more knowledge as the result of the training? Were the participants satisfied with the delivery methods and contents presented? Did the participants feel that they were competent enough to design and offer similar courses in software and information security in their home departments and institutions?

The paper is structured as followings: The importance of teaching security related concepts is emphasized in Section II. The experience with curriculum design is discussed in Section III. Section IV reviews the experiences and education challenges with two and four-year colleges. We sketch the goals of the training workshop in Section V. The methodology employed for implementing the training workshop is presented in Section VI. Section VII reports the evaluation of the project. The valuable lessons learned through offering the training workshop is listed in Section VIII. Section IX concludes this paper.

II. SOFTWARE SECURITY EDUCATION

Educational institutes have critical and key roles in supplying the needed workforce for the nation's needs. The digital world, where we all live in, needs reliable technologies to facilitate daily businesses. It is important that the educational institutes take into account the paradigmatic shift that adjusts the existing stress from *students as customers* to the *society as customers* and thus feel more responsible for addressing the needs for highly prepared security experts.

Taylor et al. [6] argue that computer security and its education is often overlooked in undergraduate degree programs. The educators are in common agreement and believe that undergraduate students must be engaged in security related areas earlier in their programs and courses such as CS0 and CS1 courses. Frazier et al. [7] propose designing and integrating security modules in both undergraduate and graduate levels and integrating software safety systems into the Computer Science program. More specifically, Frazier et al. suggest encompassing operating system security, software security testing, code review, risk and threat analysis, and database security into the CS programs, where the course module includes lecture materials, in class demonstrations as well as hands-on assignments.

Taylor et al. [8] aim to review the previous recommendations regarding the security concepts covered in Computer Science education, and highlight the significance and key role of proper and realistic tests and hands-on experiences when teaching secure coding and programming. They [8] address the four myths as widely discussed and argued when teaching secure programming and coding practices:

- There are no rooms in the curriculum for a course in secure programming. The students taking more advanced classes often focus on class materials and not on the programming styles. There is a gap between theory and what has being taught in classrooms and practices.
- If students learn how to write secure programs, the state of software and system security will dramatically improve.
- Academic institutions are hierarchical educationally. The teaching strategies and ideas are being developed through different ways. However, it is important to remember there may exist no best way.
- We know what to do and how to do it. However, often instructors do not know how to teach secure programming.

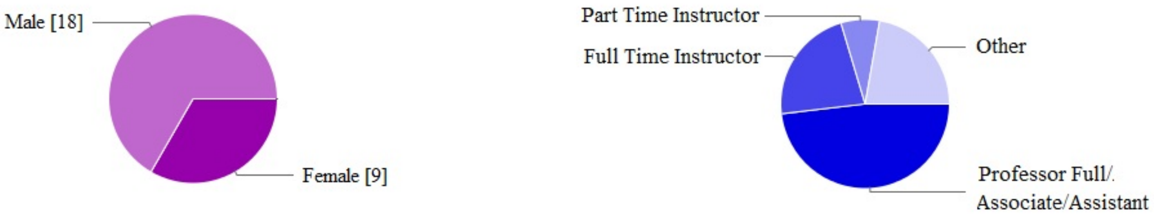
The Summit on Education in Secure Software, jointly sponsored by the National Science Foundation and Education and Human Resources (EHR), aims at developing a comprehensive agenda focusing on the formidable challenges of secure software education [9]. To meet the goals, the summit has three major objectives:

- to take into account the ideas and opinions of all cyber security stakeholders from academia, government, industry, certification and training institutions on teaching secure programming;
- to implement the ideas and opinions discussed by the cybersecurity stakeholders to create or improve existing teaching methods;
- to outline and observe the quality of education and to enable it to reach a broader audience, and shape inclusive agenda for secure software education that have objectives for dissimilar viewers, teaching methods, resources needed, and problems that are predicted to rise.

Despite its importance roles, security concepts are less discussed in most Computer Science and Software Engineering textbooks often written briefly and primarily for the undergraduate level. It is crucial to include fundamental security concepts and techniques early in the undergraduate program and more specifically into software engineering courses [10].

III. DEVELOPING AND INTEGRATING CYBERSECURITY CURRICULUM

Curriculum and course development is a practice to understand and address the student needs through designing course modules and assessing the learning outcomes. According to Conklin [11], a typical set of information security courses includes security principles, cryptography, digital forensics,



(a) Gender distribution.

(b) Career status.

Fig. 1. Gender distribution and career status of the participants attending the training workshop.

network security, application security, e-commerce security, policy and law, authentication and biometrics, and intrusion detection. Consistent with Conklin, Davis and Dark describe the common body of knowledge for computer security curricula in which four focus areas are identified as cryptography, secure computing systems, network security, and management [12].

Research on cybersecurity education shows that the current practice of cybersecurity education is unsatisfactory and needs substantial improvement to meet the required standards [13], [14], [15], [16], [17], [18], [19]. As a step forwards improving security education, Taylor and Azadegan propose using security checklists and scorecards in teaching students to develop secure code [20]. Furthermore, Taylor and Azadegan [21] present a platform for a "security across the curriculum" effort and integrating security curriculum into the core courses.

IV. Experiences with two- and four-year Institutions

Perez et al. [22] investigated two- and four-year institutions and observed that the challenges known to the research intensive educational institutions in teaching security related courses were common with these teaching institutes. Associate degree programs in information assurance, usually offered by community and technical colleges, play a key role in educating IA technicians, practitioners, and professionals in the United States. Perez et al. further claim that the large variation in the type of degree programs at community and colleges is primarily due to two factors: a) the absence of a set of curricular guidelines around which institutions can build their programs, and b) the lack of a clear understanding of the needs of the employers of the graduates of these programs. Perez et al. further suggest that developing closer and more functional relationships is needed to make progress on the articulation challenges.

As a project supported by National Science Foundation, the ACM committee for Computing Educational Community Colleges (`www.acmccecc.org`) has conducted the strategic summit on the Computing Education Challenges at Community Colleges and observed that active collaborations were often missing, which are required among the various sectors of education including two- and four-year colleges and universities. Moreover, the unique characteristics of computing education in community colleges required approaches and solutions specifically tailored to address their needs [23].

V. Goals of the Cyber Security Training Workshop

The training workshops aim at building a capacity for cyber security and thus introducing an educational model to be adopted by other geographical regions and academic institutions. The grant goals of this workshop are three-fold:

1) Offer summer professional development workshop on security related areas with its focus on software and information security for community college instructors and further assist them in developing and enhancing their own security-related course modules and curricula;
2) Design a series of follow-up activities for the participating community college instructors and mentor them for the objective of transferring the knowledge they have acquired through the training sessions to their home departments;
3) Evaluate the proposed project towards assessing its impact on capacity building of cybersecurity professionals across the southwest region and assess its applicability to similar faculty development program with the intention of introducing it as an education model to build a capacity of software and information security scholars;

This two-year faculty development project aims to increase the number of higher education instructors who can design, develop, and teach courses pertinent to software and information security related areas. The workshop also provides opportunities for faculty to update and learn various themes of cybersecurity issues related.

VI. Software Security Training Workshop

We report the workshop and sessions' structure, recruitment procedure, delivery methods, and data collection methodology.

A. *Recruitment and Participants*

The workshop organizers identified over 50 community colleges along with the two and four-year Universities located across the southwest region of the United States. The faculty members and instructors from these institutions were directly contacted and invited to apply for attending the training workshop. In the end, 27 faculty and instructors filled out the online application form and applied for the training workshop.

The applicants had a very diverse background. Figure 1(a) demonstrates the gender distribution of the applicants.

Figure 1(b) demonstrates the applicants' appointments and career status at their home institutes. The applicants were part/full-time instructors with majority holding professorship positions at their home institutes. The bar-chart given in Figure 2(a) demonstrates the ethnical background of the applicants with majority being white. Furthermore, the bar-chart given in Figure 2(b) illustrates the highest degrees the applicants' institutions offer.

The workshop organizers selected 16 applicants for the final program. The selection criteria were developed and designed in accordance with the primary goal of the workshop in promoting software and information security education across the southwest region.

B. Training Sessions and Delivery Format

The one-week long workshop was organized into themes and sessions related to software and information security. Each topic was organized into four sessions. The topics and sessions were delivered by faculty with expertise in Computer Science and Electrical Engineering. Each theme concluded with a class activity where the participants developed a course syllabus and an application plan with the intention of integrating the topics and course modules presented in each theme into their own curriculum at their home institute. Tables I and II list the topics that were presented in software security and information security training themes.

C. Data Collection

In the beginning of the workshop, an information session was delivered. The participants, who were willing to participate in the questionnaires, were asked to enter a pseudonym when completing surveys and forms for pre/post evaluation instruments. To gather participants' knowledge and skill level of curriculum development related to software security courses, a pre-test survey was administrated in the beginning of each theme. In the survey, participants were requested to provide a self-report indicating their prior knowledge in software and information security using a 5-point scale for various course modules. Similarly, the project investigators and trainers collected participants' knowledge and skill level of curriculum development related to software security courses and topics in the end of the theme and assessed the overall satisfaction/evaluation of the whole theme. An application plan was also distributed among the participants to indicate whether they had any intention or plan to transfer the knowledge gained through the workshop to their home institutions.

TABLE I. TRAINING SESSIONS FOR SOFTWARE SECURITY THEME.

Session One *Basic Concepts*	1. The Root and Cost of Software Failure
	2. Characteristics of Good Requirements (e.g. Security Requirements)
	3. Characteristics of Secure Software
	4. Secure Software Development Life Cycle
	Phase I. Security guidelines, rules, and regulations,
	Phase II. Security requirements,
	Phase III. Architectural and design reviews/threats modeling,
	Phase VI. Secure coding,
	Phase V. Black/white box testing,
	Phase IV. Determine exploitability
	5. Hands-on Experience and Case Studies on Vulnerabilities (e.g. Buffer Overflow Attacks, SQL Injection Attacks, Cross-Site Scripting, Cookie Tampering, etc.)
Session Two	1. Secure and Resilient Software Development (Attack pattern and Surface, Software security practice, System Design and Threats Categorizations and Ranking)
Advanced Topics	2. Security Testing basic processes and steps
	3. Security testing vs. traditional software testing
	4. Static Analysis (Bug findings, Style Checking, Type Checking, Security Vulnerability)
	5. Application Security Principles and Practices (use positive security models, fail security, run with least privilege, keep security simple, detect intrusions, security test infrastructure, security test services, establish secure default)
	6. Misuse and attack use case modeling
	7. Threats and Risk modeling
	8. Security design patterns
Session Three *Research Trends*	1. Risk-based Software Security
	2. Metrics and Models for Security Maturity
	3. Vulnerability management
	4. Environment hardening
	5. SAMM: Software Assurance Maturity Model
	6. Building Security in Maturity Model (BSIMM)
	7. BSI Software Security Framework
	8. Penetration Testing
	9. Web Applications and Session Attacks
	10. Security engineering
	11. Attack Graphs
	12. Model-based Risk Quantification (Risk quantification, Estimated Exposure Degrees, Component Dependency Graph, Estimated Component Vulnerability, Likelihood Estimate, Severity Analysis).

TABLE II. TRAINING SESSIONS FOR INFORMATION AND DATA SECURITY THEME.

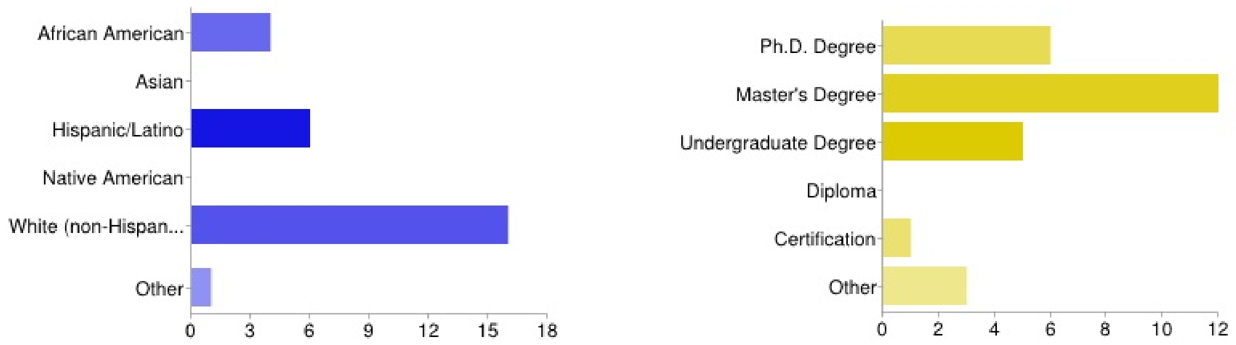
Session One *Basic Concepts*	1. Basic Network Security
	2. Basic Cryptography
	3. Conventional Cryptography
	4. Cryptography - Type of Attacks
	5. Shift Cipher
	6. Substitution Cipher
	7. Permutation Cipher
	8. Hill Cipher
Session Two *Advanced Topic I*	1. Stream Cipher
	2. Autokey Cipher
	3. Data Encryption Standard (DES)
	4. DES Modes and Triple DES
	5. Advanced Encryption Standard (AES)
Session Three *Advanced Topics II*	1. Public Key Infrastructure (PKI)
	2. Hash Functions
	3. MD5 and SHA Hash Functions
	4. Information Authentication - Digital Signature
	5. Message Authentication Code
	6. Key Distributions and Exchange

VII. EVALUATION

An education expert, the second author of this paper, designed and conducted the evaluation procedure of the workshop. The chief purpose of the evaluation was to assess whether traditional training workshops were affective in stimulating the participants' interest in designing security related courses at their home institutes.

A. The Evaluation Purpose and Questions

During the training workshop, the participants were requested to complete an anonymous survey and take part in one-to-one interviews with the project evaluator regarding their experience with the program. The major goal of the evaluation was to determine the degree of effectiveness of the traditional training workshop as well as the participants' satisfaction. The



(a) Ethnics.

(b) Highest degree.

Fig. 2. The highest degree and ethnical background of the workshop applicants.

project's impact on participants' learning outcomes was also of salient interest. The workshop evaluation was structured around the following major questions:

1) How the workshop has influenced the participants learning outcomes?
2) Do the participants know more about software and information security as a result of the training workshop?
3) Are the participants more confident in their security knowledge after the workshop?
4) How significant is the impact of the training workshop on the participants' interest?

B. Pre/Post Sample Tests

Tables III and IV list the set of questions that were asked from the participants at the beginning and end of the training sessions. Our data showed that the participants performed 50% better on the post-test questions after attending the training sessions.

C. Data Collection Instruments

The workshop organizers utilized a variety of data collection instruments and strategies (e.g. session logs, questionnaires, interviews, pre/post tests) in order to collect quantitative and qualitative data. The assessment metrics developed were as followings:

- *Knowledge*, a 5-point likert scaling metric to record the participants' prior and post knowledge related to the materials presented in each session.
- *Confidence*, a 5-point scaling metric to measure the participants' confidence for designing, developing, and offering similar software security courses at their home institutes.
- *Workshop Impact*, A 3-point scaling metric to assess the participants' opinion about the project's impact on their interests, knowledge, and confidence with security related topics.

In addition to questionnaires, a series of one-to-one interviews were also conducted to have a better insight of the participants' reflections and experiences with the training workshop.

TABLE III. PRE AND POST TESTS FOR SOFTWARE SECURITY THEME.

Pre-Test Questions	How does software fail? What is vulnerability? Give a short description of Confidentiality, Integrity, and Availability. Describe functional and non-functional requirements. List five examples of non-functional requirements. What is "misuse use case?" What is Secure Development Life Cycle (SDLC)? What is a threat model? What is security design review? List five cyber threats. How vulnerabilities get into all software? What are design and implementation vulnerabilities?
Post-Test Questions	List three security features that should be implemented when designing a banking system. Give a short description of how threats are ranked. What is attack surface? Give a short description of Physical Security, Network Security, Host Security, and Data Security. What is "user positive security model?" What is the "elevated privilege attack?" List five security practices that need to be considered when designing software. List two design phase recommendation for addressing the security of software. What is "attack use case?" What is penetration testing? How vulnerabilities get into all software? What are design and implementation vulnerabilities? What could affect the size of attack surface? What is risk-based security testing?

TABLE IV. PRE AND POST TESTS FOR INFORMATION AND DATA SECURITY THEME.

Pre-Test Questions	What is buffer overrun? What are cookies and what is cookie tampering? What is integer overflow? What is SQL injection attack? What is symmetric key? Describe Caesar cipher. What is block cipher?
Post-Test Questions	What is cross-site scripting attack? What is input validation? What is session attack? Give a short description of integrity and availability. What is Public Key Infrastructure (PKI)? What is stream cipher? What is the use of one-way hash functions in cryptography?

D. Project Impact on Participant Outcomes

We report the project impact based on the assessment metrics we developed.

TABLE V. KNOWLEDGE GAINED: PAIRED *t*-TEST RESULTS WITH 16 PARTICIPANTS (N=16).

	Pre-Workshop		Post-Workshop		*t*	*p*	Effect
Theme	**Mean**	**SD**	**Mean**	**SD**	**Value**	**Value**	**Size**
Information and Data Security	3.13	0.96	3.88	0.89	3.87	0.002	0.81
Software Security	2.63	0.96	3.38	.89	3.87	.002	0.81
Average	2.88	0.96	3.63	0.89	3.87	0.002	0.81

Knowledge Gained. The pre/post-tests scores and the result of the survey were used to assess whether the participants gained any additional knowledge through attending the training sessions. Applying the paired statistical *t*-tests, we observed that the post security knowledge of the participants was significantly improved after attending workshop. Table V lists the results of the paired *t*-tests. Furthermore, a follow up examination indicated that participants' knowledge gains were significant. Effect sizes were high, i.e. 0.81, showing that the significant differences were meaningful.

Confidence Improvement. The significance of confidence improvement was analyzed by comparing the participants' pre/post confidence scores. The results of the paired statistical *t*-tests indicated that participants' confidence significantly had an increase after attending the training workshop. Table VI reports the confidence level of the participants before and after attending the training workshop.

As Table VI indicates the effect sizes are high ranging from 0.69 to 0.82 indicating that the differences were significant and meaningful.

The Project Impact. The overall effectiveness of the training workshop was assessed by the three measurements that asked participants to rate the workshop's impact on their interest. Table VII reports the participants' opinion regarding the workshop's impact on their knowledge.

The results showed that about two-thirds of the participants strongly agreed the workshop made a significant impact on their knowledge, confidence, and interests.

VIII. LESSONS LEARNED

The evaluation data indicated that the training workshop stimulated the participants' knowledge, confidence, and interest pertinent to software and information security. While conducting surveys and questionnaires we learned that the training workshop could be further improved in order to address its goals better.

A. Peer-to-Peer Interaction

The participants expressed their satisfaction regarding meeting peers and building a network and community of scholars, who share common interests. Many of the participants have already started contacting their peers for the purpose of collaboration and sharing experiences. The participants showed their interest in learning more about their peers and their experiences in a more formal approach.

Designing and implementing relevant activities for formal peer interaction and networking before, during, and after the workshop, where the participants can share their experiences, would promote better networking, collaborations, and participant satisfaction.

B. Building A Learning Community

While the training workshop offered an opportunity where the participants could interact directly with their peers, the continuous interactions and its sustainability were also a salient factor. Building a learning community where participants could establish connections and/or enhance already existing collaborations is very crucial for successful implementation of the workshop and its goals. A well designed online platform would allow participants to build a community of practitioners and scholars and a place for exchanging ideas, products, course modules, and lesson examples.

C. Tailoring Contents to Participants' Needs

Although the evaluation data demonstrated the participants' satisfaction with the course materials presented in each session, the participants showed their interest in some course modules more than some other parts. For instance, the course module on cryptography was very exciting for some participants but not very interesting for some others. Some participants even expressed their concerns and frustration with the difficulty level of the content and materials presented in the training sessions. With respect to the feedback the workshop organizers received, it seems it would be better to conduct a preliminary assessment and gather information about the participants' interest in each related topic so that the workshop and its content could be tailored to the participants' needs and thus improve the relevance and usefulness of the workshop.

IX. CONCLUSION

The results suggest that cybersecurity and in particular software security training program promoted participating faculty's knowledge, confidence, and interest in a positive direction. However, several strategies can be integrated in future training offerings to improve participants' learning and their future activities to engage in design, development and implementation of software security courses and programs. Nonetheless, the training workshop demonstrated that it is possible to support current software security education programs at the community college level to produce both immediate graduates to fill ever-increasing vacancies, and future student cadres for higher education programs that develop administrators and scientists in the field.

ACKNOWLEDGMENT

This project has been supported by National Science Foundation under grant award DUE-SFS-1241756 to Texas Tech University.

REFERENCES

[1] C. Nickell, L. C. Prez, B. Oldfield, J. B. A. Gencer, E. Hawthorne, K. Klee, and A. L. S. Wetzel, "Towards information assurance (ia) curricular guidelines," in *In Proceedings of the 2010 ITiCSE working group reports*, 2010.

TABLE VI. Confidence level: Paired t-test with 16 participants (N=16).

	Pre-Workshop		Post-Workshop		t	p	Effect
Description	Mean	SD	Mean	SD	Value	Value	Size
I feel confident that I can successfully teach a cybersecurity course	3.63	1.20	4.44	.73	2.93	.010	0.82
I feel confident that I can create curriculum materials for a cybersecurity course	3.63	1.09	4.31	.87	2.30	.036	0.69
I feel confident that I can apply the information presented in the workshop to my teaching	4.19	.91	4.25	.93	0.27	.791	NA
I feel confident that I have the necessary knowledge and skills to design a new cybersecurity course	3.38	1.36	4.25	.93	2.91	.011	0.75
I feel confident that I can help students when they have difficulty with cybersecurity related topics	3.63	1.31	4.44	.63	2.78	.014	0.79
Average	3.69	1.05	4.34	0.65	2.61	.020	0.74

TABLE VII. Workshop impact on participants' interests, knowledge, and confidence.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I am more interested to teach cybersecurity as a result of the training	–	6.3	18.8	6.3	68.8
I am more confident in my cybersecurity knowledge as a result of the training	–	12.5	–	18.8	68.8
I know more about cybersecurity as a result of the training	6.3	6.3	6.3	18.8	62.5

[2] L. J. Hoffman, T. Rosenberg, R. Dodge, and D. Ragsdale, "Exploring a national cybersecurity exercise for universities," *IEEE Security and Privacy*, vol. 3, no. 5, pp. 27–33, 2005.

[3] M. Dark, "A profile of information security training needs on university campuses," in *In EduCause Mid-Atlantic Regional Conference Proceedings*, Baltimore, MD, 2001, pp. 1–22.

[4] "United state computer emergency readiness team (us-cert)," http://www.us-cert.gov/, August 2010.

[5] "Global security statistics and trends (trustware)," http://www.trustware.com/, 2011.

[6] B. Taylor, H. Hochheiser, S. Azadegan, and M. O'Leary, "Cross-site security integration: Preliminary experiences across curricula and institutions," in *Proceedings of the 13th Colloquium for Information Systems Security Education*, Seattle, WA, June 2009.

[7] A. Frazier, X. Yuan, Y. Li, and S. Hudson, "Course modules for software security," in *Proceedings of the 12th Colloquium for Information Systems Security Education*, 2008.

[8] B. Taylor, M. Bishop, E. Hawthorne, and K. Nance, "Teaching secure coding: The myths and the realities," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '13. New York, NY, USA: ACM, 2013, pp. 281–282.

[9] D. L. Burley and M. Bishop, "Summit on education in secure software," The George Washington University and University of California, Davis, Tech. Rep. GW-CSPRI-2011-7 and UCD-CSE-2011-15, 2011.

[10] A. Wang, "Security testing in software engineering courses," in *34th Annual Frontiers in Education*, October 2004.

[11] W. Conklin, "The design of an information security practicum course," in *Proceedings of the 2007 International Academy for Information Management - (SIGED pre-ICIS)*, Montreal, Canada, December 2007.

[12] M. Dark and J. Davis, "Defining a curriculum framework in information assurance and security," in *Proceedings of the 2003 American Society for Engineering Education Annual Conference and Exposition*, Nashville, TN, 2003, pp. 1–6.

[13] S. Cooper, C. Nickell, L. C. Pérez, B. Oldfield, J. Brynielsson, A. G. Gökce, E. K. Hawthorne, K. J. Klee, A. Lawrence, and S. Wetzel, "Towards information assurance (ia) curricular guidelines," in *Proceedings of the 2010 ITiCSE working group reports*, ser. ITiCSE-WGR '10. New York, NY, USA: ACM, 2010, pp. 49–64.

[14] B. Taylor, S. Kaza, S. Azadegan, M. O'Leary, and C. Turner, "Injecting security in the curriculum – experiences in effective dissemination and assessment design," in *Proceedings of the 14th Colloquium for Information Systems Security Education*, Baltimore, MD, 2010.

[15] M. Dark and M. Bishop, "Evaluating the efficacy of software security curriculum exercises," in *Faculty Workshop on Secure Software Development*, Orlando, FL, 2008.

[16] M. Dark and J. Davis, "Report on information assurance curriculum development," in *Colloquium for Information Systems Security Education Proceedings*, Redmond, WA, 2002, pp. 1–16.

[17] J. Ekstrom, M. Dark, and B. Lunt, "Implementation of information assurance and security in existing it curricula," in *Proceedings of the American Society for Engineering Education*, Chicago, IL, 2006, pp. 1–7.

[18] B. Bogolea and K. Wijekumar, "Information security curriculum creation: a case study," in *Proceedings of the 1st annual conference on Information security curriculum development*, ser. InfoSecCD '04. New York, NY, USA: ACM, 2004, pp. 59–65.

[19] M. Dark, J. Ekstrom, and B. Lund, "Integration of information assurance and security into the it2005 model curriculum," in *Proceedings of the ACM SIGITE 2005 Conference*, Newark, NJ, 2005, pp. 7–14.

[20] B. Taylor and S. Azadegan, "Using security checklists and scorecards in cs curriculum," in *Proceedings of the 11th Colloquium for Information Systems Security Education*, Boston, Massachusetts, June 2007.

[21] ——, "Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum," in *Proceedings of the 3rd annual conference on Information security curriculum development*, ser. InfoSecCD '06. New York, NY, USA: ACM, 2006, pp. 24–29.

[22] L. C. Pérez, S. Cooper, E. K. Hawthorne, S. Wetzel, J. Brynielsson, A. G. Gökce, J. Impagliazzo, Y. Khmelevsky, K. Klee, M. Leary, A. Philips, N. Pohlmann, B. Taylor, and S. Upadhyaya, "Information assurance education in two- and four-year institutions," in *Proceedings of the 16th annual conference reports on Innovation and technology in computer science education - working group reports*, ser. ITiCSE-WGR '11. New York, NY, USA: ACM, 2011, pp. 39–53.

[23] E. K. Hawthorne, K. J. Klee, and R. D. Campbell, "Findings from an acm strategic summit on computing education in community colleges," in *ITiCSE*, 2011, p. 373.

The Changing Nature of Teaching and Learning Shaped by Information and Communications Technology

G. Dzawo[1], M. Mbodila[2], and M. Kikunga [1]
[1]Computer Science & Information Systems, University of Venda, South Africa.
[2]Computer Science Foundation Unit, University of Venda, South Africa.

Abstract – *Information and Communications Technology (ICT) has changed the face of higher education and the nature of learning. In the 21st century, education systems have witnessed the rapid growth of information available making it imperative that students are equipped with the right skills to make choices on relevant information. Most governments today aim to offer the most comprehensive and effective education possible for their citizens within the constraints of existing funding and limited resources by increasing access to higher education. This has brought about some teaching and learning challenges some of which can be redressed by integrating ICT in the learning processes. Learning with ICT can potentially develop transferable and lifelong skills. In this paper, we highlight teaching and learning challenges faced by most higher education students in South Africa and then propose a conceptual framework for integrating ICT in teaching and learning underpinned by the constructivist theory of learning.*

Keywords: Higher education, constructivism, ICT,

1 Introduction

In today's globalised society, Information and Communications Technology (ICT) has become one of the basic building blocks of modern society. Research has shown that ICT can provide a useful context for co-operation and collaboration in educational settings between learners, between educators, and between learners and educators. Embedding ICT in educational settings diversifies the learning opportunities provided for learners. The use of ICT can lead to changes in work and changes in the way students learn. In the global economy, knowledge production has become market driven, competitive and diversified [1]. The focus of teaching and learning is now towards generic transferable skills which makes graduates more employable and simultaneously cultivating lifelong skills [1]. Learning with ICT can potentially lead to the development of such skills.

ICTs are a "diverse set of technological tools and resources used to communicate, and to create, disseminate, store, and manage information"[2]. These technologies include *computers, the Internet, broadcasting technologies (radio and television), and telephony.*

ICT as the driver of knowledge economies can develop higher order cognitive skills which are metacognition, problem solving, collaboration and learning how to learn [3]. Knowledge societies demands that lifelong skills be part of graduate attributes.

2 Background

Post-apartheid South Africa has witnessed increased demand for participation in higher education but that has also brought about many challenges, amongst them the high attrition rate and low throughput[4]. According to the National Plan for Higher Education (NPHE) produced in 2001 by the Department of Higher education, South Africa had a graduation rate of 15% which is one of the lowest in comparison with other countries the world over [5]. According to a study by the Council on Higher Education[6] , a large number of students who enter university drop out before they finish their studies. In most higher education institutions, fewer students are graduating in regulation time and many drop out before graduating. Some of those that manage to graduate do not have adequate graduate attributes that are required by the society and industry.

This high dropout rate is one of the factors that contribute to the critical shortage of skilled manpower in the country [5]. The skills shortage includes key sectors of the economy such as ICT. Scott [4] also reports that the requirements of skilled staff in the country are not being met by the graduate output. Funding for higher education institutions is related to graduate throughput forcing universities to enrol more students with inadequate infrastructure. Scott [4] noted that just increasing enrolment numbers without matching with the effectiveness of the educational process compromise standards and quality of graduates.

In rural universities in South Africa, most students come from poor backgrounds and some of them enrol on a part-time basis.

3 Challenges related to teaching and learning

The transition period from high school to university impacts on student retention and non-completion [7]. Although factors like lack of funds, wrong choice of programme and unrealistic expectations contribute to their learning challenges, the major one is that students do not have the required type of skills and rigour of learning and understanding expected at university [7]. It is during the transition period that teachers make students aware that learning at university requires critical thinking and application of knowledge at higher cognitive levels than they have previously been used to [7]. This involves changing perceptions, learning and study approaches and adopting new epistemological beliefs.

Academic preparedness has been cited as one of the major challenges facing higher education in South Africa [8]. Students are entering university underprepared, and some of them use the surface approach to learning. Most students come to university with inadequate reading, writing and study skills [9]. Other challenges related to teaching and learning language issues since English is a second language to must students, large classes and inadequate curriculum design [8].

To address the challenge of low throughput and high attrition rates in South African Universities, students must be taught independent learning skills. Scott [4] argues that some factors that affect students' performance may be out of control of higher education but higher education must take the responsibility to address those factors within its control.

In the face of all these challenges, it is imperative that efforts be directed to harness the potential and immense contributions that can be derived from integrating ICT in higher education curriculum. The proliferation of digital portable and mobile devices makes it more attractive to take advantage of available technology to ease and improve learning.

4 Benefits of using ICT in teaching and learning

Learning with ICT and not learning from provides a useful conceptual framework when integrating ICT into teaching and learning [10]. The use of ICT in education in general, helps students to enhance their performance on basic skills and provides a rich learning environment that allows them to search for information. Cuttance [11] argues that many people in schools or higher education institutions assume that exposure to technology will lead to adequate learning about how to use it for improving learning. In the past decade the research focus on ICT use in general has been shifting away from access, infrastructure and technical issues towards use of ICT to enhance learning [12].

With ICT, students can learn anytime, anywhere. One defining feature of ICT is their ability to transcend time and space [2]. ICT make possible asynchronous learning, or learning characterized by a time lag between the delivery of instruction and its reception by learners.

Technology facilitated learning now allows off-campus delivery of instruction for students who cannot attend classes for some other reasons. Students can access remote learning resources through learning management systems such as Moodle.

Several factors are now making the adoption of ICT in education more attractive. These include the influence of pervasive use of the Internet and the emergence of ubiquitous computing technologies.

ICT enabled education has been categorised into three main classes, e-learning, blended learning and distance learning [13]. E-learning or electronic learning is the use of electronic media and information and communication technologies (ICT) in education. This includes all forms of educational technology in learning and teaching. Some of many advantages e-learning include eliminating time and geographical boundaries for leaners and teachers and enabling student interaction and collaboration [13]. Distance learning uses information technologies to deliver instruction to learners who are at remote in locations. It provides the basis for e-learning. Blended learning combines face-to-face learning , self-paced learning and online learning [13].

Oliver [13] highlights a number of factors that have inhibited the adoption of ICT in education which are lack of funding to purchase the required equipment, lack of expertise on the part of academics and a general lack of motivation.

Various stakeholders can benefit from ICT in a variety of ways as summarised in the Table 1.

Table 1: Benefits of ICT to the main stakeholders.

Stakeholder	Benefits
Student	• Increased access, • Flexibility of content and delivery, • Combination of work and education, • Learner-centred approach, • Higher-quality of education and new ways of interaction.
Employees	• High quality, cost effective professional development in the workplace, • Upgrading of employee skills, increased productivity, • Developing of a new learning culture, • Sharing of costs and of training time with the employees, • Increased portability of training.
Governments	• Increase the capacity and cost effectiveness of education and training systems, • To reach target groups with limited access to conventional education and training, • To support and enhance the quality and relevance of existing educational structures, • To ensure the connection of educational institutions and curricula to the emerging networks and information resources, • To promote innovation and opportunities for lifelong learning.

Source UNESCO (2002) [14]

4.1 The nature of learning

Learning is described as “the acquisition of knowledge, competences, skills and attitudes that leads to long term changes in behavioural patterns, cognitive structures, or personal identity” [15]. Learning theories are define as, “ formulations of apparent relationships or underlying principles that have been verified to some degree” [15]. We briefly review the learning theories in pedagogical practice that underpin our proposed framework.

4.1.1 Constructivist theories of learning

The constructivism theory of learning, asserts that students constructs their knowledge instead of simply receiving it from the teacher [16]. Constructivism is a philosophical, epistemological, and pedagogical approach to learning [17]. An important aspect of constructivism described by Ben-Ari (2001) is that knowledge is constructed and understood based on the student’s past knowledge and social setting. Thus people understand the things that they construct themselves and reflecting on their experiences in a social context where there is collaboration with their peers. Realistic problem-solving situations enable students to be actively engaged in constructing knowledge.

One of the characteristics of constructivism is that knowledge is acquired recursively, that is, reflecting on existing knowledge to create new cognitive structures which are again used to create knew knowledge[16]. Constructivism shares a lot with discovery or inquiry learning, in which students have to discover knowledge by themselves in a given scenario [16]. However, reading books and lecturing remain indispensable.

There are two types of constructivism, cognitive constructivism in which learning is considered as an active process where students build new ideas using prior knowledge and social constructivism which emphasises that intelligence grows out of culture ([17], [18]). According to Ben Ari [16] the main goal of constructivism is metacognition, which is the higher order process of reflecting on our own thinking and problem solving processes.

Application of the constructivism theory in the computer science discipline is not as widespread as in other science disciplines [16]. Ben Ari [16] argues that constructivist approaches can be applied thorough all the stages of computer science education. Wulf [19] discusses the constructivist approach to the teaching of programming, Connolly and Begg [17] reports on a constructivist-based approach to teaching database design and analysis.

4.1.2 Connectivism

Siemens [21] describes another learning theory called connectivism. The interconnectedness of work experience, learning and knowledge including the influence of technology on our everyday life are the critical aspects of this theory. According to Siemens [21] learning and knowledge exist in diversity of opinions. Learning is regarded as a process of connecting specialized nodes or information sources. The information that exists in non-human appliances is a form of knowledge that needs to be nurtured to facilitate continual learning [21]. Digital knowledge is knowledge that can be processed and transformed by ICT.

.

5 ICT enabled learning framework

The proposed framework in Figure 1, describes the components of an ICT integrated learning environment based on the constructivist theory of learning. Our framework is informed by other theories drawn from the literature; Phillips etal's Learning Environment, Learning Processes and Learning Outcomes (LEPO) framework [22], Biggs' Presage, Process, Product (3 – P) model [23], and Laurillard's conversational framework[24].

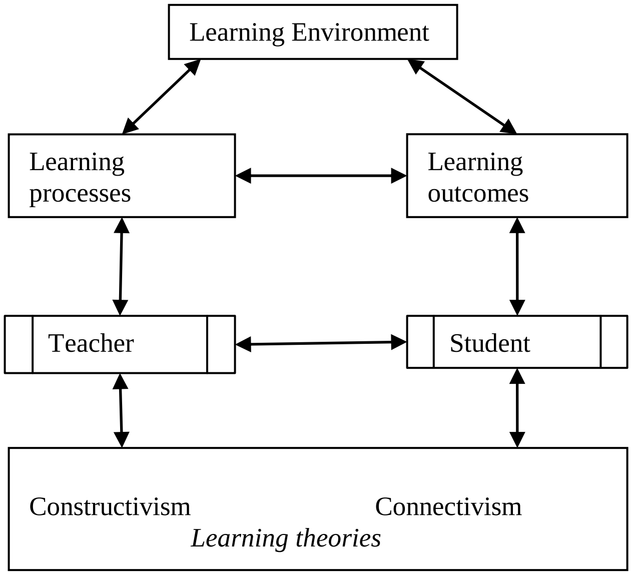


Figure 1. ICT integrated learning framework

5.1 The role of the Teacher

The teacher provides coaching and scaffolding so that students can use their existing knowledge to acquire new knowledge. Scaffolding is the support given to students such as hints and constructive feedback[17]. Providing constructive feedback after an assessment tasks helps students be aware of their progress and to identify areas they need to take remedial action. The teacher designs the constructive learning environment to facilitate learning. The teacher facilitate the establishment of knowledge, that is to enable students to think and behave like discipline specialists. Teaching enables participation in knowing. The role of the teacher is to give the student the capacity to understand the unfamiliar discourse.

To bring about the conceptual change which is the goal of education, the teacher needs to adopt a student-focused teaching strategy [20]. Constructive alignment is an approach to curriculum design that integrates all the elements of teaching and learning [20]. The teacher should ensure that all components in the teaching system, that includes, purpose, intended outcomes of the course, teaching methods applied, and the assessment tasks are aligned to each other.

5.2 The role of the student

In student centered learning, students have control over their learning. With the constructivist theory, the students are actively engaged in the construction of their knowledge. In promoting active learning, ICT enhanced learning provides technologies for examination, calculation and analysis of information, thus providing a platform for student inquiry, analysis and construction of new information [2]. Learners learn by doing when engaged on appropriate real-life problems. In contrast to memorization-based or rote learning, ICT enhanced learning promotes increased learner engagement.

5.3 The learning environment

This is an ICT enabled constructivist learning environment, which has the entire elements that support learning, that is, the curriculum, the university facilities, physical and online resources.

5.4 The learning outcomes

Learning outcomes are what the students are able to do as a result of engaging in a course, that is, after going through the learning experience.

5.5 The learning processes

5.5.1 Collaboration

Collaboration involves learners working together to generate deeper levels of understanding towards a common goal sharing and constructing a certain level of common knowledge, understanding and expertise[25]. Various names are used for collaborative learning: cooperative learning, collective learning, learning communities, peer teaching, peer learning, or team learning [25]. Important skills of peer interaction, communication, and group work are developed through collaboration.

Collaboration between students leads to a better understanding of the learning activities and facilitates sharing and distribution of knowledge and expertise. The use of ICT changes the nature of students learning by giving them an opportunity to collaborate. Learning management systems such as Moodle have facilities for online discussion forums where students can share learning experiences. Skype and WhatsApp are some of the tools that students can use. It is the responsibility of higher education institutions to train lifelong skills in students so that they can engage in self-directed learning in informal settings [3]. Internet sources like Wikipedia can facilitate informal learning.

5.5.2 Social

Learning is influenced by the environment, society and developed through social interaction with others. Social constructivism emphasises that human intelligence grows out of culture [18]. ICT provides a communication platform for sharing knowledge. Social interaction on networking sites such as Facebook can lead to independent learning

5.5.3 Authentic

The ability to solve problems in real authentic environments is an important attribute that must be developed in students. ICT allows us to simulate such environments.

5.5.4 Reflective

ICT have the potential to develop metacognition skills of self-reflection and critical thinking.

6 Conclusion

The ICT teaching and learning framework support constructivist learning approaches for lifelong learning, metacognition skills of self-reflection and critical thinking, complex problem solving and collaboration.

The changing nature of higher education, coupled with the recent technological advancements is transforming the role of the teacher from being a transmitter of knowledge to that of a facilitator of learning. Teachers need to develop new and innovative teaching strategies that encourage critical thinking in students. These strategies include encouraging independent and collaborative learning based on the epistemological perspective that students construct their own knowledge in a social context. ICT enhanced learning promotes increased learner engagement.

7 References

[1] K. Luckett, “A proposal for an epistemically diverse curriculum for South African higher education in the 21st century,” *South African J. High. Educ.*, vol. 15, no. 2, pp. 49–61, 2001.

[2] V. Tinio, *ICT in Education*. New York,NY 10017: Stephen Browne, 2003.

[3] K. W. Lai, “Digital technology and the culture of teaching and learning in higher education,” *Australas. J. Educ. Technol.*, vol. 27, no. 8, pp. 1263–1275, Dec. 2011.

[4] I. Scott, “First-year experience as terrain of failure or platform for development? Critical choices for higher education,” *Focus first-year success Perspect. Emerg. from South Africa beyond*, pp. 17–36, 2009.

[5] M. Letseka and S. Maile, *High university drop-out rates: A threat to South Africa’s future*. Human Sciences Research Council Pretoria, 2008.

[6] S. A. Council on Higher Education, “A proposal for undergraduate curriculum reform in South Africa: The case for a flexible curriculum structure,” 2013.

[7] U. Wingate, “A framework for transition: Supporting ‘learning to learn’in higher education,” *High. Educ. Q.*, vol. 61, no. 3, pp. 391–405, 2007.

[8] S. Jaffer, D. Ng’ambi, and L. Czerniewicz, “The role of ICTs in higher education in South Africa: One strategy for addressing teaching and learning challenges.,” *Int. J. Educ. Dev. using Inf. Commun. Technol.*, 2007.

[9] S. Brüssow, “No Title,” *A Learn. Facil. Framew. to Enhanc. Acad. Ski. Dev. among underprepared Learn. South African High. Educ.*, 2007.

[10] L. Tay, S. Lim, C. Lim, and J. Koh, “Pedagogical approaches for ICT integration into primary school English and mathematics: A Singapore case study,” *Australas. J. …*, 2012.

[11] P. Cuttance, “School innovation: Pathway to the knowledge society,” *Commonw. Aust.*, 2001.

[12] J. Blackmore, L. Hardcastle, E. Bamblett, and J. Owens, *Effective use of information and communication technology (ICT) to enhance learning for disadvantaged school students*. 2003.

[13] R. Oliver, “The role of ICT in higher education for the 21st century: ICT as a change agent for education,” *Retrieved April*, 2002.

[14] UNESCO, “Information and Communication Technologies in Teacher education: A Planning Guide,” 2002.

[15] E. Rosado and C. Bélisle, “Analysing digital literacy frameworks,” *… /13/77/79/PDF/Analysing-Edu-Frameworks. …*, 2006. [Online]. Available: http://lire.ish-lyon.cnrs.fr/IMG/pdf/Analysing-Edu-Frameworks.pdf. [Accessed: 31-May-2014].

[16] M. Ben-Ari, “Constructivism in computer science education,” *J. Comput. Math. Sci. Teach.*, vol. 20, no. 1, pp. 45–73, 2001.

[17] T. M. Connolly and C. E. Begg, “A constructivist-based approach to teaching database analysis and design,” *J. Inf. Syst. Educ.*, vol. 17, no. 1, p. 43, 2006.

[18] J. S. Atherton, “Learning and Teaching; Constructivism in learning,” 2011. [Online].

Available: http://www.learningandteaching.info/learning/constructivism.htm. [Accessed: 20-Jun-2013].

[19] T. Wulf, “Constructivist approaches for teaching computer programming,” *Proc. 6th Conf. Inf. Technol. Educ. (pp. 245-248)*, 2005.

[20] J. Biggs, “Aligning the curriculum to promote good learning,” in *Constructive Alignment in Action: Imaginative Curriculum Symposium, Learning and Teaching Support Network*, 2002.

[21] G. Siemens, “Connectivism: A learning theory for the digital age,” *Int. J. Instr. Technol. Distance Learn.*, vol. 2, no. 1, pp. 3–10, 2005.

[22] R. Phillips, C. McNaught, and G. Kennedy, “Towards a generalised conceptual framework for learning: the Learning Environment, Learning Processes and Learning Outcomes (LEPO) framework,” 2010.

[23] J. Biggs, “Aligning teaching for constructing learning,” *High. Educ. Acad. 8 August 2007].Available from http//www.heacademy.ac.uk/assets/York/documents/resources/resourcedatabase/id477_aligning_teaching_for_constructing_learning.pdf*, 2003.

[24] D. Laurillard, “Rethinking university teaching: A conversational framework for the effective use of learning technologies New York: Rutledge,” 2002.

[25] M. Dooly, “Constructing knowledge together,” *A Guideb. to moderating Intercult. Collab. online*, 2008.

Decimal to Binary Number Conversion can be Fun

Irene Polycarpou[1]
[1]School of Sciences, University of Central Lancashire Cyprus, Larnaka, Cyprus

Abstract *- Numbering systems are of great importance in Computer Science and Engineering education. The binary numbering system can be considered as one of the most fundamental, since its understanding is essential for the understanding of other Computer Science and Engineering concepts, such as data representation, data storage, computer architecture, networking, and many more. Yet, students are having difficulties understanding it. One approach which has been shown to improve learning of different science and mathematics concepts is the use of educational games. Educational games have the potential to engage and motivate learners through fun activities. This paper presents a small exploratory survey on an electronic educational game for practicing decimal to binary number conversions.*

Keywords: undergraduate Computer Science and Engineering education; binary numbers; number systems; conversion of decimal to binary numbers; educational games.

1 Introduction

This paper builds on [16], which presents an educational game for practicing decimal to binary number conversions, called Binary Blaster. Numbering systems provide the foundation for many Computer Science and Engineering topics, making them one of the most fundamental concepts in Computer Science and Engineering education [13]. Of significant importance is the binary numbering system, which is widely used in Computer Science and Engineering. Understanding of the binary system is essential for the understanding of other Computer Science and Engineering topics, including but not limited to, data storage, computer architecture, and networking [32]. Moreover, given that at their most basic level, computer data (including computer's instructions), are represented with binary numbers (without reducing those numbers further to simple high and low voltages), the binary number system is essential for understanding data representation. Also, specific representations such as twos-complement and floating point, build on and require an understanding of binary numbers [17].

The IEEE/ACM Joint Task Force on Computing Curricula guidelines for undergraduate degree programs include machine level representation of data (machine learning) as part of one of the fundamental knowledge areas. Some of the recommended topics to be covered are numeric data representation, number bases, and twos-complement representations [8]. Computer Science and Engineering students should have an understanding of the binary number system so that they can understand the basic operations of a machine. At even a more general level, to be able to develop useful and efficient software and hardware, students should have a good understanding of the process involved in binary arithmetic and operations performed by computers.

Although the binary numbering system is at the core of many areas of Computer Science and Engineering education, students have difficulties understanding it and instructors explaining it [17, 23]. This may be an inherent difficulty, since it is an unfamiliar concept for the students. Most students are so used to performing arithmetic in the decimal system that any other numbering system seems unnatural [1] and difficult for them to comprehend [12]. Moreover, according to Petzold (2000), conversions between the common number systems can be tricky and awkward [7].

One approach that has been shown to be effective in teaching and learning different science and mathematics concepts is the use of educational technologies. By bringing the worlds of academia and technology together, we can create educational environments that were not previously possible. Technology can change the way educators and students interact and communicate with each other, the way learning materials are presented, and the place and time learning occurs [15]. Successful technology integration in education can capitalize on students' enthusiasm about technology [31] and serve as a motivation for learning.

Educational software and specifically, electronic educational games, is one kind of technology that can be used to support education even for young students [2, 11, 18, 24, 30]. Research has shown that electronic games are engaging and effective learning tools [3, 10, 22]. They can motivate, excite, and engage students in the learning process. This does not come as a surprise, taking into consideration the many hours students from very young age spent playing electronic games [25]. Moreover, games can provide an active learning environment. Actively learning, making mistakes, and learning from them, is more effective than passively listening to a lecture [29].

Educational games can be designed and developed as fun and engaging activities where students can learn or practice different concepts. Games focused on learning, incorporate information which allows students to construct knowledge during game play through the exploration of different scenarios related to the target subject area. On the other hand, games focused on practice rather than learning, incorporate questions directly related to the relevant subject area and provide students incentives to answer the questions to the best of their knowledge by either offering resources necessary to continue playing the game, moving up a skill level, or earning points.

Currently, there are many electronic resources available for teaching and learning the binary number system as well as conversions between decimal and binary numbers [4, 5, 6, 9, 14, 20, 21, 26, 27, 28, 32]. Such resources can be classified into four main categories: (a) resources which provide instructional material in a static textual and visual form [5, 9, 26], (b) resources which provide instructional material only in visual form (e.g., videos) [21], (c) resources which provide a decimal to binary converter without any instructional material [4, 20, 27, 28], and (d) resources which are only focusing on practicing number conversion [6]. A more comprehensive discussion on educational software available for binary number systems can be found in [32].

Of interest to this paper is the fourth category which includes educational software for practicing number conversions and especially, educational games. In what follows, we provide an overview of an educational game, called Binary Blaster [16], which is designed for practicing decimal to binary number conversion. We also present an exploratory survey on students' perceptions about the game.

2 The Game

2.1 Game Overview

Binary Blaster allows students to practice decimal to binary number conversion in three different forms of binary representation (straight binary, excess, and twos-complement). The game includes two playing modes (single player mode and multi-player mode) and two possible input modes (standard keyboard and video game controller).

When the game begins, players must select the binary representation they would like to practice with as well as the number of bits they would like the binary numbers to have (e.g, 3-bit, 4-bit, etc. up to 8-bit binary numbers).

During game play, a randomly generated decimal number embedded in a colored shape (e.g., star, circle, etc.) is "falling" from the top of the screen. At the same time, at the bottom of the screen there is a list of distinct binary numbers (the default size of the list is six distinct binary numbers) with a uniform number of bits, including the binary number that corresponds to the "falling" decimal number. The form of the binary numbers available depends on the player's selection at the beginning of the game (i.e., binary representation and number of bits). The goal of the player is to match the "falling" decimal number with its corresponding binary number available at the bottom of the screen (see, Figure 1).

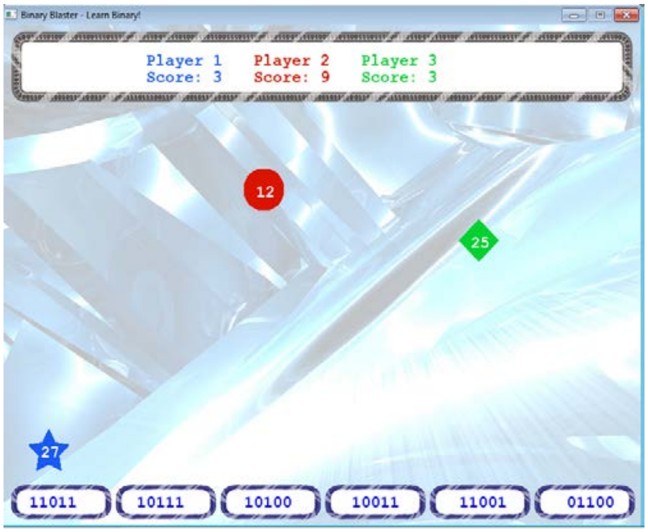


Figure 1: A screenshot from the multi-player mode of Binary Blaster showing three different players, represented by three differently colored shapes.

Once the decimal number reaches the bottom (i.e., it is matched with a binary number), immediate visual and auditory feedback is provided to the players. To motivate players to spend more time practicing decimal to binary conversion so as to become proficient, players receive points for every correct matching and lose points for every incorrect one. The player's goal is to earn points and achieve a high score. Following feedback to the user, a new decimal number starts falling from the top of the screen and the list of available binary numbers is updated. The entire process is repeated until the players have reached a desired score or feel that they are proficient with the specific type of conversion.

Moreover, the game provides different ways to challenge the players:

(a) the speed of the falling decimal number increases as the players progress through the game and earn more points;
(b) players can increase the number of bits in the binary numbers they are practicing with;
(c) players can increase the number of available binary numbers to choose from (at the bottom of the screen);
(d) players can change the binary number representation (i.e., standard binary, excess notation, or twos-complement).

Binary Blaster is currently available for different operating systems (e.g., Linux, Windows, iOS) and although it only includes conversions from decimal to binary representation, it is easily expandable to other numbering systems such as hexadecimal and octal.

A more detailed discussion of the game and its implementation can be found in [16].

3 Exploratory Survey

3.1 Overview

To further explore the potential of Binary Blaster as well as confirm and strengthen the findings of [16], a small exploratory survey on its effectiveness and students' perception of it, was conducted with first year undergraduate computing students. The survey was part of the "Introduction to Mathematics for Computing" course, in which students are introduced to numbering systems, with emphasis on the binary system. All students taking the course (40 students) were invited to participate in the survey voluntarily. A total of 38 students accepted the invitation. Although the number of participants is not very high (i.e., the survey results may not be representative), the survey results can be used to indicate future directions.

During the course, one class period (1 hour and 15 minutes) was spent on formal instruction of the binary system and binary representations. By the end of this class period, participants were introduced to Binary Blaster and a one hour extra session was arranged for participants to play with the game. The session was scheduled two days after formal instruction and it took place in a computing laboratory. During the "play" session students were asked to use the game individually to practice decimal to binary conversions. Some students chose to play in groups (multiplayer mode) in addition to playing individually. Students were also asked to try and play the game with different number of binary bits and different binary representations, until they felt comfortable with the conversions.

3.2 Survey

Three instruments were developed for evaluating the effectiveness of the game as well as students' perceptions about the game: the pre-test worksheet, the post-test worksheet, and the post-test questionnaire. The pre-test worksheet was administered to the students at the beginning of the game play session. The post-test worksheet and post-test questionnaire were administered at the end of the game play session. Below are the three instruments.

Pre-test Worksheet:

Problem 1:
Convert the following binary numbers to decimal. On a scale of 1 (very easy) to 5 (very difficult) rate the degree of difficulty for each.
a) 110 b) 011 c) 1001 d) 1011
e) 11100 f) 10101 g) 101100 h) 110100

Problem 2:
Convert the following decimal numbers to binary. On a scale of 1 (very easy) to 5 (very difficult) rate the degree of difficulty for each.
a) 7 b) 4 c) 13 d) 10
e) 20 f) 29 g) 42 h) 45

Post-test Worksheet:

Problem 1:
Convert the following binary numbers to decimal. On a scale of 1 (very easy) to 5 (very difficult) rate the degree of difficulty for each.
a) 101 b) 100 c) 1101 d) 1100
e) 11010 f) 10011 g) 101010 h) 110011

Problem 2:
Convert the following decimal numbers to binary. On a scale of 1 (very easy) to 5 (very difficult) rate the degree of difficulty for each.
a) 6 b) 3 c) 14 d) 11
e) 18 f) 27 g) 52 h) 46

Post-test Questionnaire

Question 1: On a scale of 1 (very difficult) to 5 (very easy), how easy was Binary Blaster to use?

Question 2: On a scale of 1 (not enjoyable) to 5 (very enjoyable), how much did you enjoy playing Binary Blaster?

Question 3: On a scale of 1 (not effective) to 5 (very effective), how effective do you think Binary Blaster was for your understanding of binary numbers?

Question 4: Is there something you would like to improve about Binary Blaster? Explain your answer.

Question 5: Do you have any additional comments related to your experience with Binary Blaster?

3.3 Results

Results of participants' performance in the pre- and post-test worksheets are presented in Table 1. The table presents the percentage of participants' correct responses to the conversions (percentages are rounded to the nearest whole number) and the average participants' difficulty rating for each conversion (rounded to the nearest tenth).

As we can see from Table 1, the results are not very surprising. Participants' performance was improved after using the game, but not significantly. Moreover, according to participants' responses, their perceived difficulty with the conversions was improved, especially with conversions of 5 and 6 digit binary numbers (problem 1e-h) and large decimal numbers (problem 2e-h).

Table 1. Pre-test and post-test results

	% Correct Responses			Rating (1-5)		
	Pretest	Posttest	Change	Pretest	Posttest	Change
1a	90	90	0	1.3	1.3	0
1b	79	79	0	1.3	1.2	-0.1
1c	82	90	+8	2.4	2.2	-0.2
1d	71	79	+8	2.6	2.4	-0.2
1e	61	79	+18	3.2	2.7	-0.5
1f	68	79	+11	3.7	3.0	-0.7
1g	61	79	+18	4.0	3.4	-0.6
1h	71	90	+19	4.2	3.2	-1.0
2a	87	90	+3	1.3	1.3	0
2b	90	90	0	1.4	1.2	-0.2
2c	79	90	+11	1.9	1.5	-0.4
2d	68	68	0	2.3	1.9	-0.4
2e	71	79	+8	3.1	2.5	-0.6
2f	68	79	+11	3.4	2.7	-0.7
2g	61	79	+18	4.0	3.2	-0.8
2h	61	68	+7	4.1	3.4	-0.7

In addition to comparing participants' performance results between the pre-test and post-test worksheets, the time participants spent on each worksheet was also recorded. The average time to complete the pre-test worksheet was 18 minutes, whereas for the post-test worksheet was 11 minutes. This suggests that the game helped participants to become faster when performing number conversions (i.e., there was an improvement on participants' response time). This was also reflected on participants' comments on Question 5 of the post-test questionnaire. More than half of the participants commented that the game helped them "think faster" and become quicker while performing the conversions. Some of their responses include:

-"The game helps you think faster."

- "While playing the game you need to become quicker in doing the conversions in your mind. After playing the game for a while I became faster."

- "The game helps you find the conversions faster, you become quicker."

- "The game challenges you and helps you think fast about the conversions."

In terms of participants' feedback about the game, the results of the participants' responses to the post-test questionnaire suggest that the game was easy to use, enjoyable, and assumed to be effective by the participants. For Question 1, the average participants' response was 4.8 (i.e., participants found the game very easy to use), for Question 2 was 4.6 (i.e., participants found the game very enjoyable), and for Question 3 was 4.2 (i.e., participants' found the game effective for their understanding of binary numbers). In terms of Question 4, almost all participants responded that there was nothing they would like to change about the game because they liked the game and they felt that the game was appropriate for its purpose. Only five participants made suggestions and those were as follows:

- "I did not like the colors much. The interface is too pink for me."

- "It will be nice if the game includes some instructions on how to do the conversions rather than being just for practicing."

- "Maybe allow the players to change difficulty level while paying instead of having to exit the game and start again."

- "Include more numbers."

- "The user should be able to terminate the game when makes 3 mistakes without losing the high score."

For Question 5, the main "themes" of participants' responses were that the game was interesting and fun to play, it was challenging, and motivate them to practice more compared to a regular textbook. Some of the participants' comments are presented below:

- "It was effective for my learning because I practiced more, since I was playing to break high score."

- "I liked the game a lot. It is good and fun. The game became more effective for my learning when it was going faster. It helps your mind think fast."

- "I liked it because it reminded me of old arcade games. It was effective for learning, especially the faster it gets the more challenging and interesting it gets."

- "I liked the fact that the binary numbers were changing positions every time. It was difficult to memorize."

- "It is good that you have multiple choice. It helped me realize my mistakes and fix them."

- "The game is perfect for its purpose. I had fun playing with it. The multiplayer mode is more fun. I will use it at home also."

4 Conclusion and Discussion

One approach that has been shown to be valuable for enhancing teaching and learning, is effectively utilizing educational technologies, or more precisely, interactive multimedia and educational software in the classroom. In the last few decades, there has been a dramatic increase in the use of educational technologies available for teaching and learning different science concepts and some universities have already integrated such systems into their curricula. In this paper, an exploratory survey on the effectiveness and students' perceptions on an electronic educational game for practicing decimal to binary number conversions, called Binary Blaster, was presented. The game aims to provide a fun activity for the students which can motivate them to practice number conversions and become proficient in the subject. Although the game currently only includes decimal to binary conversions, it is easily expandable to other numbering systems such as octal and hexadecimal. The game is suitable for any age especially, high school students and first year university students.

Overall, the results of the survey suggest that the game is easy to use, enjoyable, and students' perceive it as effective for their understanding of the binary system. Moreover, it was found that students became faster in performing decimal to binary conversions after playing the game for some time. This is not a surprise, since extensive practice always results in better performances. On the other hand, according to students' comments, they only had extensive practice because of the game. In other words, the game was successful in motivating students to solve more problems and play until they felt they were proficient with the conversions. In terms of performance, there was an improvement after students played with the game, but not significant. Finally, after using the game, students found conversions easier to perform, especially with large numbers (e.g., with five and six digit binary numbers). These findings confirm and strengthen the results of [16] that students enjoy playing with Binary Blaster, the game is easy to use and fun to play with, and that in their opinion, the game is effective for their learning and understanding of binary number conversions.

For the future, a more comprehensive study will be conducted on the effectiveness of the game as well as its appeal to the students.

5 References

[1] A. D. Patterson, L. J. Hennessy. “Arithmetic for Computers,” *Computer Organization and Design: The Hardware/Software Interface.* Morgan Kaufmann. Publishers. 2007.

[2] A. Ireland, D. Kaufman, & L. Sauvé. Simulation and Advanced Gaming Environments (SAGE) for Learning. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education,* pp. 2028-2036, 2006. Reeves, T. & Yamashita, S., Eds. Cheasapeake, VA.

[3] A. Waraich, A. “Using Narrative as a Motivating Device to Teach Binary Arithmetic and Logic Gates,” in the *Proceedings of the ITiCSE'04 Conference,* pp. 97-101, 2004.

[4] Base Converter. Available from http://www.cut-the-knot.org/binary.shtml, 16 April, 2014.

[5] B. Hanselton. “Bases.” 2007 Available from http://www.brainjammer.com/math/bases/, 16 April, 2014.

[6] CISCO Learning Network Games. Available from https://learningnetwork.cisco.com/docs/DOC-1803, 16 April, 2014.

[7] C. Petzold. “Alternatives to Ten. Code: The Hidden Language of Computer Hardware and Software,” *Microsoft Press*, 2000.

[8] CSC2013. “Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.” *The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery*. Available from: http://www.acm.org/education/CS2013-final-report.pdf, 16 April, 2014.

[9] CSTC: Computer Science Teaching Center. “Conversion Between Different number Systems.” Available from http://courses.cs.vt.edu/~cs1104/number_conversion/convexp.html, 16 April, 2014.

[10] D. Oblinger. “Simulations, games and learning,” *EDUCAUSE Learning Initiative,* pp. 1-6, 2006.

[11] G. G. Bitter and M. J. Legacy. “Using Technology in the Classroom (7th ed.). Pearson Education, Inc. USA, 2008.

[12] G. J. Brookshear. “Data Storage. Computer Science: An Overview.” Pearson Education, Addison-Wesley, 2009.

[13] G. R. Dattatreya. “A Systematic Approach to Teaching Binary Arithmetic in a First Course.” *IEEE Transactions on Education,* 36 (1), pp. 163-167, 1993.

[14] H. Mitsuhara, H. Ogata, K. Kanenishi, and Y. Yoneo. “Do Children Understand Binary Numbers by Electric Card Game?” *in the first IEEE International Workshop on Digital Game and Intelligent Toy Enhances Learning (DIGITEL'07),* pp. 191-192, 2007.

[15] I. Polycarpou. “Using Technology to Enhance K-12 Outreach in Materials Science.” *Materials Research Society (MRS) Bulletin,* 36 (4), pp. 290-297, 2011.

[16] I. Polycarpou, J. Krause, and M. Noring. “Binary Blaster: An Educational Game for Practicing Binary Number Conversions,” in the *Proceedings of the Int'l Conf. Frontiers in Education: CS and CE (FECS'11)*, pp. 492-497, 2011.

[17] K. Becker. “A Multiple Intelligences Approach to Teaching Number Systems,” in *Proceedings of 2003 ITiCSE conference*, pp. 250, 2003.

[18] K. Squire & H. Jenkins. “Harnessing the Power of Games in Education.” *InSight*, 3(5). Institute for the Advancement of Emerging Technologies in Education (IAETE). 2003.

[19] L. A. Steen. “Twenty Questions about Mathematical Reasoning,” in L. Stiff (ed.), *Developing Mathematical Reasoning in Grades K-12*, pp. 270 – 285, 1999. Reston, VA: National Council of Teachers of Mathematics.

[20] Math is Fun. Available from http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html, 16 April, 2014.

[21] Math Warehouse. Available from http://www.mathwarehouse.com/non-decimal-bases/convert-binary-to-decimal.php, 16 April, 2014.

[22] M. Mayo. “Games for science and engineering education,” *Communications of the ACM*, 50 (7), 2007.

[23] M. Panoiu, A. Iordan, C. Panoiu, L. Ghiorghioni. “Educational Software for Teaching the Basics of Computer Science,” *WSEAS Transactions on Advances in Engineering Education*, 12 (6), pp. 238-243, 2009.

[24] M. Prensky. “Digital game-based Learning.” McGraw-Hill, NY., 2001.

[25] N. Etuk. “Educational Gaming: From Edutainment to Bona Fide 21st Century Teaching Tool.” Available from http://www.mmischools.com/Articles/ReadArticle.aspx?ArticleID=59693, 16 April, 2014.

[26] Numbering Systems Tutorial. Available from http://www.electronics.dit.ie/staff/tscarff/number_systems/number_systems.html, 16 April, 2014.

[27] Numbers Converter. Available from http://www.unitconversion.org/unit_converter/numbers.html, 16 April, 2014.

[28] On-line Binary Decimal Converter. Available from http://www.binaryconvert.com/, 16 April, 2014.

[29] R. Turban. “Approaches to Implementing and Teaching Human Computer Interaction,” in the *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC’03)*, pp. 81-84, 2003.

[30] R. Van Eck. “Digital game-based learning: It’s not just the digital natives who are restless.” *EDUCAUSE Review*, 41(2), pp. 16-30, 2006.

[31] V. J. Rideout, U. G. Foehr, and D. F. Roberts. “Kaiser Foundation Study: Generation M2.” Available from http://www.kff.org/entmedia/upload/8010.pdf, 16 April, 2014.

[32] Y. Feastery, F. Aliz, and J. O. Hallstromy. “Serious Toys: Teaching the Binary Number System,” in the *Proceedings of the ITiCSE’12*, pp. 262-267, 2012.

6 Acknowledgements

The author would like to thankfully acknowledge the contributions of Julie Krause, Kyle Schulz, David Danford, Monica Noring, and Heidi Lewis to the development of Binary Blaster.

Session Keys for Encryption/Decryption in Elliptic Curve Cryptosystems

Donovan Moore, Michael Gubody, and Tai-Chi Lee
Department of Computer Science and Information Systems
Saginaw Valley State University
University Center, MI 48710

Abstract - *This work evolved from a case study on an Elliptic Curve Cryptosystem (ECC) [2], where the session keys are used with FPGAs in the process of encryptions or decryptions [4]. To improve the strength of encryption and the speed of processing, the public key and the private key of ECC are used in 3BC (Block Byte Bit Cipher) [1, 5, 11] algorithm, which generates session keys for the data encryption. We are investigating a novel approach of hardware co-design implemented in VHSIC Hardware Description Language (VHDL), which produces hardware algorithm for heavy iterations to be placed onto the FPGAs, thereby gaining a speed-up by a subroutine call to a sequence of custom instructions executed on the FPGAs.*

Keywords: EEC, FPGA, VHDL, 3BC algorithm

1 Introduction

The session keys play an important role in the process of encryptions/decryptions for an ECC (Elliptic Curve Cryptosystem). The majority of products that use public-key cryptography for encryption/decryption use RSA algorithm. But as we know, the key length for secure RSA has increased over the years. This would demand a heavy computing power for applications, especially for electronic commerce site that process a large number of transaction. Recently, a different approach of generating public key based on elliptic curve cryptography (ECC) has begun to challenge the weakness of RSA [12]. Its security relies on the problem of computing logarithms on the points of an elliptic curve. The main attraction of ECC is that it appears to offer equal security for a far smaller key size, thereby saving the processing overhead. To improve the strength of encryption and the speed of processing, the public key and the private key of ECC are used in the 3BC (Block Byte Bit Cipher) algorithm, which generates session keys for the data encryption. Fundamentally, ECC (Elliptic Curve Cryptosystem) technique is more mathematics involved. We only give a brief review of the basic concept in the next section and will explain elliptic curve ciphers later.

2 EC (Elliptic Curves) - Mathematical Overview

The elliptic curve cryptosystem makes use of elliptic curve in which the variables and coefficients are all restricted to elements of a finite field. Two families of elliptic curves are used in cryptographic applications. They are prime curves defined over Zp and binary curves constructed over GF(2^n). In general, cubic equations for elliptic curve over real numbers takes the form

$$y^2 + axy + by = x^3 + c\,x^2 + d\,x + e. \quad (1)$$

where a, b, c, d, e are real numbers that satisfy some conditions. For our purpose, we will limit to the case where a, b, and c are zero, which results in the form

$$y^2 = x^3 + d\,x + e. \quad (2)$$

To plot such curve, we need to compute

$$y = (x3 + d\,x + e)^{1/2} \quad (3)$$

For given values of a and b, the plot consists of positive and negative values of y for each value of x. Thus each curve is symmetric about y = 0. Figure show two examples of elliptic curves. In the definition of an elliptic curve, we include a single point O called a point at infinity or the zero point, and also if any three points on EC lie on a straight line, their sum is O. Thus, the addition of two points on EC is defined as follows:

1) O serves as the additive identity so that
$O + O = O$, and
$P + O = P$, for any P ε EC

2)There exists an inverse for any point P on EC. The inverse of P = (x,y) on EC is
$-P = (x,-y)$, and the subtraction is defined as
Follows $P - Q = P + (-Q)$, for any P, Q ε EC

3)An associative law.
$P + (Q + P) = (P + Q) + R$, for any P, Q, R ε EC

4)A commutative law.

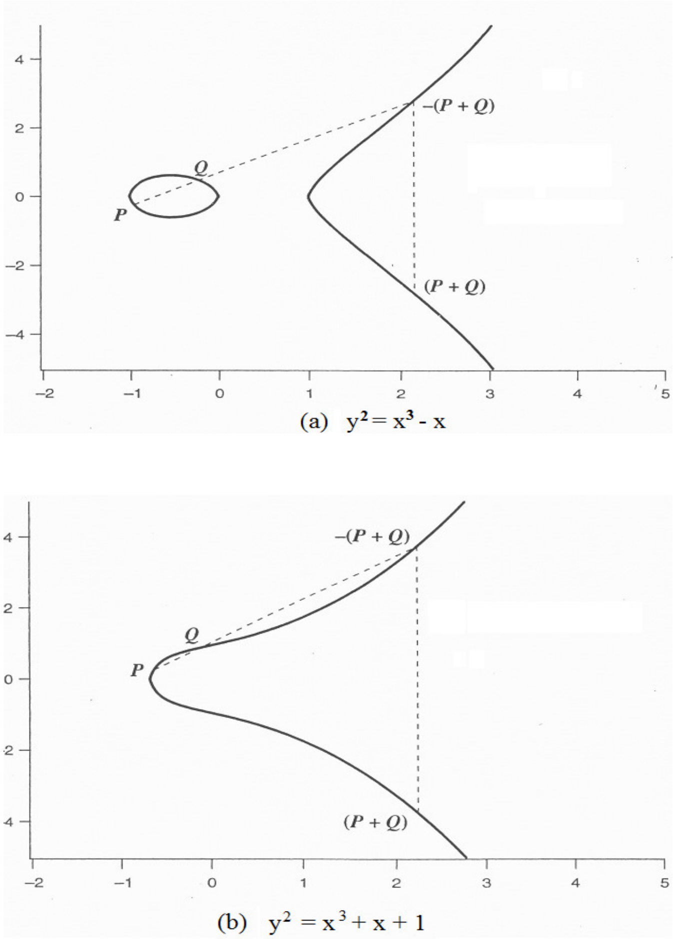
$P + Q = Q + P$, for any P, Q ε EC

For any two distinct points P = (xp, yp), Q = (xq, yq) that are not negative of each other, the slope of the line *l* that joins them is m = (yq - yp) / (xq - xp). There is exactly one other point where *l* intersects the elliptic curve, and that is the negative of the sum of and Q. After some algebraic manipulation, we have the sum R = P + Q as follows:

$$x_r = m^2 - xp – xq$$
$$y_r = - yp + m (xp – xr)$$

We also need to be able to add a point to itself, that is P + P = 2P = R. When yp ≠ 0, we have

$$xr = [(3xp^2 + a)/2yp]^2 – 2xp$$
$$yr = [(3xp^2 + a)/2yp] (xp – xr) – yp$$



(a) $y^2 = x^3 - x$

(b) $y^2 = x^3 + x + 1$

Figure 1 . Examples of Elliptic Curves

2.1 ECC (Elliptic Curve Cryptosystem)

The concept of ECC, which was proposed by N. Kobiltz [5] and V. Miller [11] in 1985 is that when any two points are selected and added, the point of the sum is generated and is used for cryptosystem. The elliptic curve (EC) over real numbers x is the set of points (x,y) to satisfy a equation y2 = x3 + ax + b. If the right side term x3 + ax + b doesn't have multiple root, that is, 4a3 + 27b2 ≠ 0, EC gives us some geometric features to work with. To apply EC to cryptosystem, the computation with modulo p is used where p is prime number. As a result of this calculation, rounding errors to be raised by real number computation can be prevented. The procedure to generate a public key in ECC is outlined as follows:

(1) [Sender] Select any prime number p
(2) [Sender] Select any integer number a, b for EC such that $y^2 = x^3 + ax + b$
(3) [Sender] Select randomly an initial point P on EC
(4) [Sender] Generates a random integer as private key K_S
(5) [Sender] Computes a public key K_S by multiplying P by K_S and registers it in the public key directory.
(6) [Sender] Transmits p, a, b, P, K_SP to Receiver
(7) [Receiver] Receives p, a, b, P, K_SP from Sender
(8) [Receiver] Generates a random integer K_R as a private key
(9) [Receiver] Computes a public key K_RP by multiplying P by K_R and registers it in the public key directory.

Note: It is easy to verify that K_S K_R P = K_R K_S P

3 Encryption and Decryption Algorithm

As shown in Figure 2, the user A computes a new key $k_A(k_BP)$ by multiplying the user B's public key by the user A's private key k_A. The user A encodes the message by using this key and then transmits this cipher text to user B. After receiving this cipher text, The user B decodes with the key $k_B(k_AP)$, which is obtained by multiplying the user A's public key, k_AP by the user B's private key, k_B. Therefore, as $k_A(k_BP) = k_B(k_AP)$, we may use these keys for the encryption and the decryption.

Select factor: a, b, $y^2 = x^3 + a x + b$.
p: prime number, P: Point on EC

User A	User B
Private key: K_A	Private key: K_B
Public key: K_AP	Public key:K_BP
en/decryption key:	en/decryption key:
$K_A(K_BP)$	$K_B(K_AP)$
↑	↑

Figure 2. Concept of en/decryption of ECC

In the next section, we investigate 3BC cipher algorithm, which is a different approach that adds an additional level of complexity to enhance the security of encryption.

3.1 Encryption and Decryption with 3BC Algorithm

With 3BC algorithm, the procedure of data encryption is divided into three parts, inputting plaintext into data block, byte-exchange between blocks, and bit-wise XOR operations between data and session key.

3.2 Session Key Generation

As we know that the value which is obtained by multiplying one's private key by the other's public key is the same as what is computed by multiplying one's public key to the other's private key. The feature of EC is known to be almost impossible to estimate a private and a public key. With this advantage and the homogeneity of the result of operations, the proposed 3BC algorithm uses a 64-bit session key to perform the encryption and decryption. Given the sender's private key Ks and the receiver's public key Pr, we multiply Pr by Ks to obtain a point KsPr = (X, Y) on EC, where X= X1 X2,…Xm and Y= Y1 Y2,…Yn. Then we form a key N by concatenating X and Y (i.e. N = X1 X2,…Xm Y1 Y2,…Yn), and generate the session keys as follows:

i) If the length (number of digits) of X or Y exceed four, then the extra digits on the left are truncated. And if the length of X or Y less than four, then they are padded with 0's on the right. This creates a number N' = X1' X2' X3' X4' Y1' Y2' Y3' Y4'. Then a new number N'' is generated by taking the modulus of each digit in N' with 8.

ii) The first session key sk1 is computed by taking bit-wise OR operation on N" with the reverse string of N".

iii) The second session key sk2 is generated by taking a circular right shift of sk1 by one bit. And repeat this operation to generate all the subsequent session keys needed until the encryption is completed. For more details on the use of public key and session key for encryption and decryption process, see [6].

3.3 Block Data Input

The block size is defined as 64 bytes. A block consists of 56 bytes for input data, 4 byte for the data block number, and 4 byte for the byte-exchange block number (see Figure 5). During the encryption, input data stream are blocked by 56 bytes. If the entire input data is less than 56 bytes, the remaining data area in the block is padded with each byte by a random character. Also, in the case where the total number of data blocks filled is odd, then additional block(s) will be added to make it even, and each of those will be filled with each byte by a random character as well. Also, a data block number in sequence is assigned and followed by a byte-exchange block number, which is either 1 or 2.

3.4 Byte Exchanges between Blocks

After filling the data into the blocks, we begin the encryption by staring with the first data block and select a block, which has the same byte-exchange block number for the byte exchange. In order to determine which byte in a block should be exchanged, we calculate its row-column position as follows:

For the two blocks whose block exchange number, n = 1, and given the values of a sender's public key 21135 and a receiver's private key 790, the row and column can be computed.

For n = 1, It follows from 3.2 that N" = 11357900 (after truncation, padding and concatenation), and

row = ((1,1,3,5,7,9,0,0)*1) mod 8 = (1,1,3,5,7,1,0,0) and

col = (((1,1,3,5,7,9,0,0)*1+3) mod 8) = (4,4,6,0,2,4,3,3)

This results 8 byte-exchange positions, (1,4), (1,4), (3,6), (5,0), (7,2), (1,4), (0,3) and (0,3). However, counting only once for repeating pairs, the four bytes at (1,4) (3,6), (5,0), and (7,2) will be selected for byte-exchange between two blocks.

For block exchange number n = 2, we have

row = ((1,1,3,5,7,1,0,0)*2) mod 8 = (2,2,6,2,6,2,0,0) and

col = (((1,1,3,5,7,1,0,0)*2+3) mod 8 = (5,5,1,5,1,5,3,3),

which results 8 byte-exchange positions, (2,5), (2,5), (6,1), (2,5), (6,1), (2,5), (0,3) and (0,3). Similarly, only three byte positions at (2,5), (6,1), and (0,3) are used for byte-exchanges between two blocks.

3.5 Bitwise XOR between Data and Session Keys

After the byte-exchange is done, the encryption proceeds with a bit-wise XOR operation on the first 8 byte data with the session sk1 and repeats the operation on every 8 bytes of the remaining data with the subsequent session keys until the data block is finished.

Note that the process of byte-exchange hides the meaning of 56 byte data, and the exchange of the data block number hides the order of data block, which needs to be assembled later on. In addition, the bit-wise XOR operation transforms a character into a meaningless one, which adds another level of complexity to deter the network hackers.

4 Theoretical Implementation

To generate a public key, the most time consuming process is to find an initial point P on the given elliptic curve and to compute kP for an integer k < p for a large prime number p. The approach we investigate in this paper is to create a 64bit ALU with its own custom instructions added to an Altera EP2C35 NIOS II embedded processor. Custom instructions are designed to be small, rearrangeable portions of a C implementation of key generation. This will allow sections of the algorithm to be in C and other sections to be expressed as custom instructions. These sections can be easily reordered and re-factored by recompiling the algorithm and uploading the overlay to the FPGA via TCP/IP in order to handle the distribution of the algorithms over the network.

4.1 Hardware Design

The design of this approach consists of four components: A PC Master Controller, TCP/IP interconnect, FGPA logic units that each contains a NIOS II processor and custom ALU, and the creation and selection of the custom instructions and overlays.

4.2 PC Master Controller

A PC Master Controller will provide benefits over existing designs. It is capable of systematically assigning algorithms to logic units based on the specific set of custom instructions included in the ALU. Our design will implement the delegation of operations and also take advantage of the parallelism that can be obtained by using a FPGA [3,7,8,10].

4.3 TCP/IP Interconnect

The PC Master Controller will communicate through a TCP/IP interface with one or more FGPAs in a cluster. Each FPGA will execute the algorithm, using the custom instructions.

4.4 Custom Instructions

The Altera EP2C35 NIOS II embedded processor is a 32bit system. By adding a customized 64bit ALU and associated 64bit registers [9, 12, 13], we can have custom instructions to handle algorithms specific to public key via EC, which include: XOR, Addition, Multiplication, Division, Right/Left Shift, and others. These instructions are given 32bit UUIDs as their opcode, allowing unique naming even when the full set is not within a single ALU. We are experimenting with various decompositions of Expansion and Permutation of the 3BC algorithm between PC Master Controller and custom instructions for optimal results.

5 Current Implementation

5.1 Implementation of 3BC Encryption

The current approach to implementing the 3BC encryption algorithm assigns the blocking and byte exchange to the PC Master Controller, with the bitwise right rotation and XOR handled on the ALU implemented on an FPGA. The current model involves using the ALU to generate one 64-bit session key from a prior session key to encrypt 64-bits of data.

5.2 Hardware Implementation

At the time of writing, the hardware model is as follows:

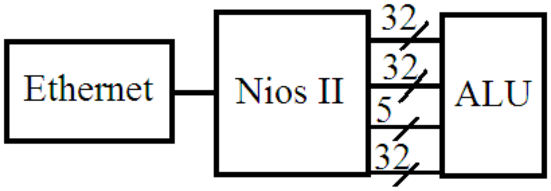


Figure 3. Current Implementation Model with Custom ALU

In figure 3, two 32-bit input buses provide key, plaintext, and the number of bits to shift the prior session key into the ALU for key generation, a 5-bit control bus for the NIOS II custom instruction, and one 32-bit output bus for the encrypted result from the ALU. The NIOS II chip handles the network connection and manages the barrel shift and XOR by means of a custom instruction with some I/O handling for data locality.

The inside of the ALU is outlined below:

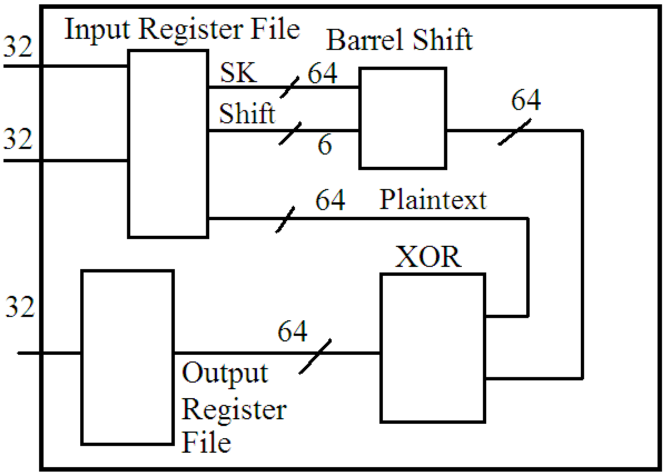


Figure 4. Inside of Custom ALU on FPGA (Not pictured: 5-bit control bus for custom instruction)

This model suffers from the limitation of requiring 3 clock cycles to input the data from the NIOS II chip to the input register file, 2 clock cycles to output the data, an assumed 1 clock cycle to process the data, as well as any inherent delays from Ethernet data transfer. Future research plans include changing to 64-bit data buses as input to the custom ALU, or a change to a different System on a Programmable Chip (SOPC) processor. With these, the input delay can be reduced to 2 clock cycles or even 1. A 64-bit data bus for output from the ALU allows for output to be condensed into 1 clock cycle, allowing for the process to be completed in an expected 3-4 clock cycles.

6 Acknowledgements

The student authors would like to acknowledge the SRCI mini-grant foundation for the opportunities it provides, SVSU for providing a prosperous research environment, and Dr. Lee for his resources and encouragements.

7 Empirical Results

Different decompositions of the 3BC sub-algorithms are being investigated. With the approach on the encryption/decryption algorithm outlined in Section 4, the expectation is to process on average one key block per clock cycle, excluding filling and extracting the custom ALU data. This appears reasonable as the custom instructions allow the design to use several FPGAs to process multiple key ranges simultaneously. Of interest is the location of the balance between the high implementation time with the low run time of the pure hardware approach, and the low implementation time with high run time of the pure software approach.

For instance, starting with a pre-developed encryption or decryption algorithm in C that has a nonexistent design time, its average run time is a constant. When the algorithm is translated into well optimized hardware, the design time is very high and the run time is very low. With the approach outlined in Section 4, the design time and run times are between the pure hardware and pure software methods. When the number of data sets to run is in the bolded range on Figure 3, this method should be preferable.

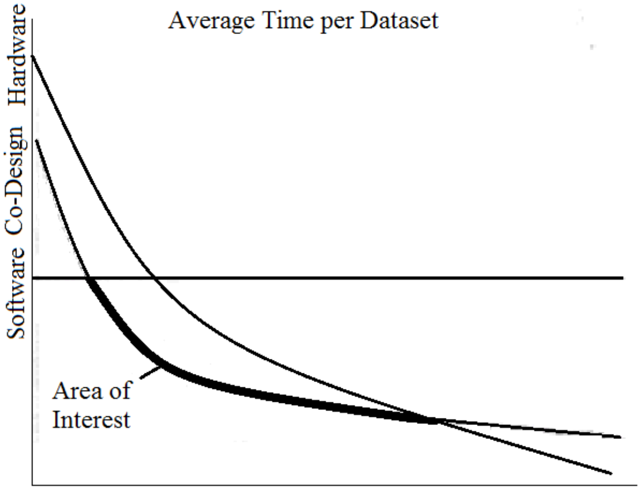


Figure 5. Area of Interest with Co-Design

7.1 Difference in Efficiency

The parallel algorithm described in section 4 allows for a large increase in speed over the pure software approach.

For pure software, the assumption is that the first session key (SK1) has already been generated, there are 6 additional clock cycles needed to generate the session keys for the first block (SK2-7), and 7 clock cycles are needed to XOR the data with session keys. Thus, a total of 14 clock cycles are required for each block after the first. For n blocks, the operations take f(n) = 14n clock cycles. This is an operation in O(n) time.

With sufficient hardware, the data can be input in 1 clock cycle. After the first block, the algorithm requires 7 operations that can be performed in parallel for session key generation. With the FPGA, this can be accomplished in 1 clock cycle. There are then 7 XOR operations between session keys and data that can also be performed in parallel. This, too, with the FPGA can be performed within 1 clock cycle. Lastly, an anticipated 1 clock cycle for gathering the results and output. Thus, for n blocks and sufficient hardware, the operations take f(n) = 4 clock cycles. This is an operation in O(1) time.

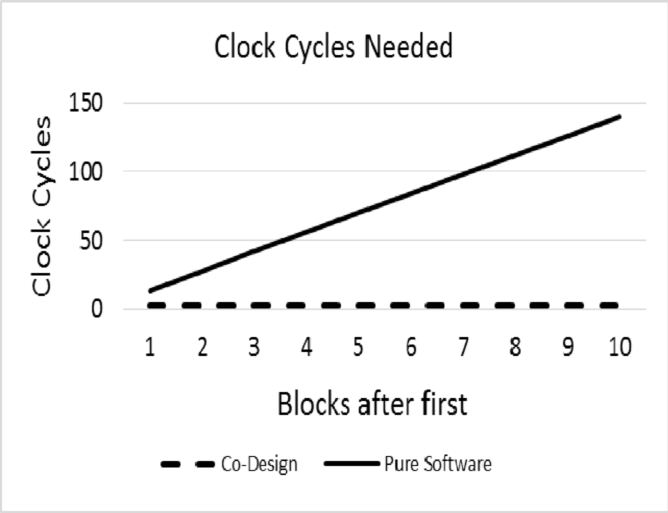


Figure 6. Theoretical Efficiency gained with sufficient hardware

8 Conclusion

The researched approach has the benefit of some of the speed of ASIC, while maintaining some of the flexibility of C. The added use of storing and transferring the algorithms as an overlay allows the organizational aspects of the algorithm to be re-factored and delivered without the need of rebuilding the ASIC image and reconfiguring the FPGA [10] Using a TCP/IP interconnect network to send both the overlay and the problem set allows for an efficient and easily scalable infrastructure.

The proposed 3BC, which uses byte-exchange and the bit operation increases data encryption speed. Even though cipher text is intercepted during transmission over the network. Because during the encryption process, the 3BC algorithm performs byte exchange between blocks, and then the plaintext is encoded through bit-wise XOR operation, it rarely has a possibility for cipher text to be decoded and has no problem to preserve a private key

9 References

1. M.J. Bastiaans, FPGA's as Cryptanalytic Tools, http://www.sps.ele.tue.nl/ members/m.j.bastiaans/spc/rouvroy.pdf.
2. A. Fernaades, Elliptic Curve Cryptography, Dr. Dobb's Journal, December, 1999.
3. P. Glesner, and M. Zipf, Renovell (Eds.): Programmable Logic and Applications. Reconfigurable Computing Is Going Mainstream, Proceedings of 12th International Conference, FPL 2002, Montpellier, France, September 2-4, 2002.
4. Jilani Ibrahim, Li Lan, Shi Yixin, VHDL implementation of Data Encryption Standard (DES), pp. 3-6, 16, ECE Department, University of Illinois at Chicago May 9th, 2003.
5. N. Koblitz, Elliptic Curve Cryptosystems. Math. Comp. 48 203-209, 1987.
6. Tai-Chi Lee, Byung Kwan Lee, An ASEP (Advanced Secure Electronic Payment) Protocol Design, The Proceedings of The IEEE 2004 International Conference e-Technology, e-Commerce and e-Service, Taipei, Taiwan, pp. 41-46, 3/28-31, 2004.
7. Tai-Chi Lee, Mark White, Using Software Emulation in FPGAs To Improve Co-Design Development Time, The Proceedings of the Fourth International Conference of Applied Mathematics and Computing, Bulgaria, pp. 359-360, August 12-18, 2007.
8. Tai-Chi Lee, Patrick Robinson, A FPGA-Based Designed for an Image Compressor, International Journal of Pure and Applied Math, Academic Publications, Volume 33 No.1, pp. 63-67, 2006.
9. Tai-Chi Lee, Richard Zeien, Adam Roach, and Patrick Robinson, DES Decoding Using FPGA and Custom Instructions, The Proceedings of The Third International Conference on Information Technology: New Generation, Las Vegas, Nevada, pp. 575-577, 2006.
10. Tai-Chi Lee, Patrick Robinson, and Erik Henne, Framework for executing VHDL code on FPGA, The Proceedings of the International MultiConference in Computer Science & Computer, Las Vegas, NV, pp. 1296-1299, 2004.
11. V.S. Miller, Use of Elliptic Curve in Cryptography. Advances in Cryptology-Proceedings of Crypto '85, Lecture Notes in Computer Science 218, pp. 417-426, Springer-Verlag, 1986.
12. M. Robshaw, Block Ciphers, RSA Laboratories Technical Report TR, 601, August 1995, http://www.rsasecurity.com/rsalabs/dindex.html.
13. B. Schneier, Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish), Proceedings, Workshop on Fast 78 Software Encryption, New York: Springer-Verlag, 1993.

SESSION

LATE BREAKING PAPERS AND POSITION PAPERS: COMPUTER SCIENCE, APPLIED COMPUTING, AND MATHEMATICS EDUCATION

Chair(s)

Prof. Hamid R. Arabnia

Collaborative Software Development: The Thomas Paine Encyclopedia

Lubomir Ivanov
Department of Computer Science, Iona College
715 North Avenue, New Rochelle, NY 10801
USA
livanov@iona.edu

***Abstract*—We outline the collaborative effort in developing the mobile application "The Thomas Paine Encyclopedia" – an educational and research tool for historians, political scientists, and 18th century literature students and experts. The project was a collaboration between a computer scientist and a political historian, whose expertise complemented each other and helped shape the application as the most detailed, historically accurate resource on Paine and his life and works. The app provides an immersive historical experience for experts and students alike.**

Keywords: Mobile App, Thomas Paine, Research/Education tool, User Interface

I. INTRODUCTION

DURING the past decade, Digital Humanities has become a very active interdisciplinary research area aiming to bridge the gap between traditional Humanities disciplines and Computer Science. The goal of this large-scale endeavor is to develop and employ computer-based methodology and software in Humanities research and education. As part of this effort, we created and deployed "The Thomas Paine Encyclopadia" – a unique mobile application devoted to the life and works of Thomas Paine.

Thomas Paine was one of the Fathers of the American Revolution, an influential 18th century political thinker, philosopher, inventor, and writer, whose fiery words ignited the American Revolution, supported the French Revolution, and continue to inspire people around the world to this day. Paine's major works - "Common Sense", "American Crisis", "Rights of Man", "Age of Reason", "Agrarian Justice" - are a required reading for many high school and college level courses in History, Political Science, and Literature.

In 2012, the Institute for Thomas Paine Studies (ITPS) - a collaboration between Iona College, NY and the Thomas Paine National Historical Association (TPNHA) – urged the development of a mobile application dedicated to the life and works of Thomas Paine. This application was intended to be an integral part of a wider educational effort, which included the development of a Thomas Paine curriculum and a research project on authorship authentication of Thomas Paine works [1]. The development of the "Thomas Paine Encyclopedia" (TPE) application was a collaborative effort between the author (a computer scientist) and the coordinator of ITPS, Gary Berton (a historian).

This paper describes the development process, the innovative features of the TPE application, and considers its international impact on Paine education and research.

II. DESIGN GOALS

We set out to design the Thomas Paine Encyclopedia as a one-of-a-kind, interactive, mobile application meeting a number of design goals:

- Provide an exhaustive and complete set of biographical, literary and other resources on Thomas Paine, his life, times, and works
- Provide the highest level of historical accuracy
- Provide an immersive user experience capturing the historical nature of the material presented
- Ensure ease of use for all application users
- Ensure usability for both research and education

To achieve these goals we had to carefully consider the availability of historically accurate materials and literary texts, as well as the software architecture most appropriate for presenting these materials in a hardware-constrained mobile environment.

III. HISTORICAL MATERIALS

The materials used in the application were methodically collected and cross-verified by Gary Berton – an expert on Thomas Paine studies, a historian for TPNHA, and the coordinator of ITPS. Mr. Berton used a large number of resources [2-12] to compile the most historically accurate biography of Thomas Paine based on facts that Paine scholars agree on and on the latest research results available. Next, the biographical material had to be scaled down to accommodate the mobile nature of the application. The material was then broken into a number of chronological pages, each devoted to a significant milestone in Paine's life.

A number of historical images and maps were also collected in support of the biographical write-up. The dates of these maps and images correspond closely to the actual dates of the events described in the biographical text. The maps and image were obtained from a variety of sources such as the TPNHA Collection, the Library of Congress, and the David Rumsey Historical Maps Collection.[1] Copyright permissions and wavers were obtained for all images and maps not in the public domain.

The literary works of Thomas Paine have been meticulously collected and archived by the TPNHA. Direct access to all of Paine's writings is available through the TPNHA website [13] or through the TPE application.

Lubomir Ivanov is with Iona College, 715 North Av. New Rochelle, NY, 10801 (email: livanov@iona.edu, phone: 914-633-2342)

[1] For a complete listing of the image/map resources, see the Acknowledgements section at the end of this paper.

IV. Software Architecture

In accordance with the goals set forth at the beginning of the project, the software architecture was designed to reflect the historical nature of the project and to provide an intuitive navigation and an easy access to the wide host of biographical and literary material in a hardware resource constrained mobile environment. The application is designed as a collection of components – Biography, Selected Works, Anti-Paine Cartoons, Quotes, and Credits – accessible through a common main interface.

A. *The Main Interface*

The main interface of the TPE application consists of a number of overlapping, semi-transparent, moving panels, which provide access to the specific application components.

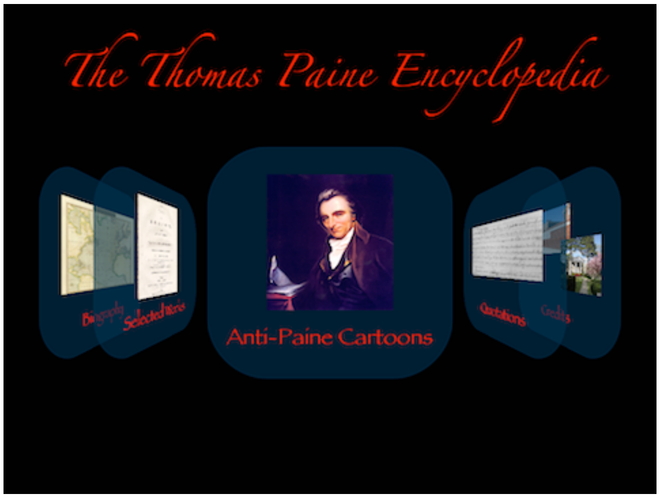


Fig.1 The main application interface

The panels are moved by swiping to the left or to the right. The active panel is in the center, and tapping on it brings forth the specific application component. The motion of the interface is controlled through a physics-based software system, which uses friction and inertia to control the scroll speed of the panels based on the length of the swipe.

To provide a uniform look-and-feel to the overall application, the main interface is mirrored at the navigation levels of each of the application's sub-components. All components except "Biography" use a similar panel-based navigation, but the individual look-and-feel is different for each component so that they can be easily differentiated based on their appearance.

B. *The Biography Component*

The Biography component is designed around a multi-level, map-based navigation system. Its purpose is to combine the chronology of events in Thomas Paine's life with the specific locations where these events took place, thus providing a more immersive, visually stimulating, and true historical experience to the users. Accurate historical maps, dated very closely to the actual time of events, were used to create the layered navigation system, which consists of three layers – "world", "country", and "city".

The top ("world") navigation level presents the user with a map of Europe, America, and the Atlantic Ocean - the locales upon which all events of Paine's life took place.

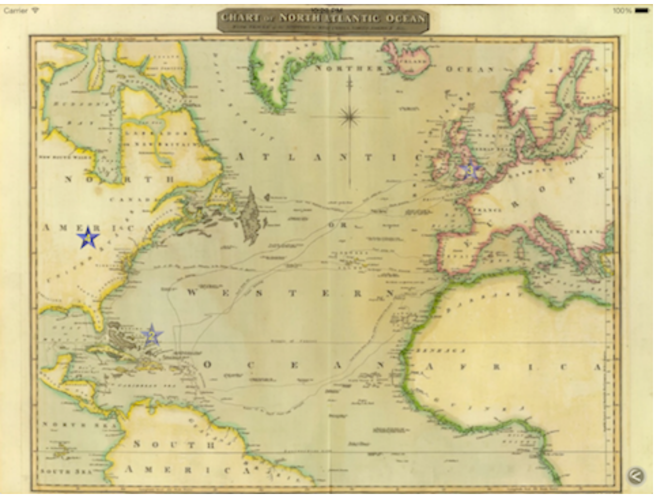


Fig.2 Top ("World") level of the Biography navigation system

The locale of the next chronological event is denoted with a blue star on the map. Locations that have already been visited are marked with a faded blue star. Tapping on an event star brings up the next level of the navigation controller – a specific country map. There are separate maps for each country that Paine lived in or visited – England, US, France, and the Caribbean.

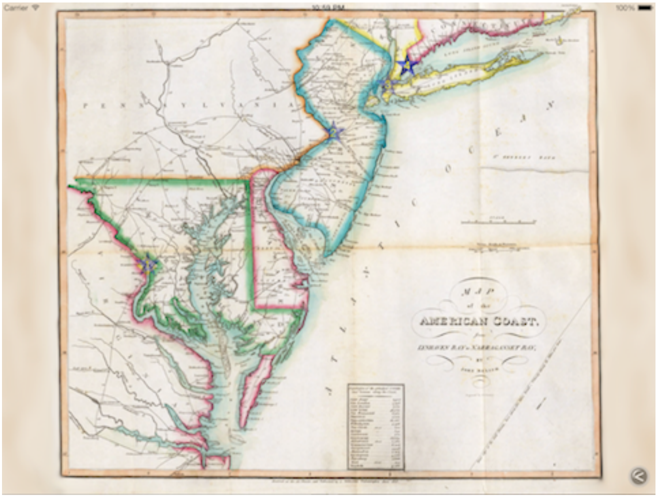
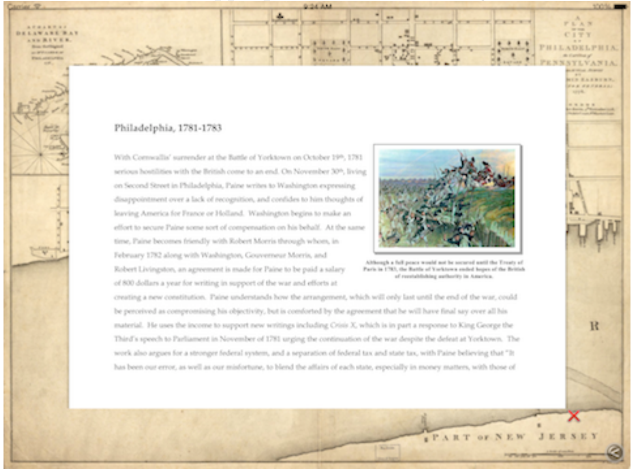


Fig.3 Second ("Country") level of the Biography navigation system

The country level of the biographical navigation interface geographically and chronologically localizes the events to a city or an area of a country denoted by a numbered blue star. One further localization occurs at the lowest level of the biographical navigation controller – the city level. At the city level, events are localized to actual physical addresses from the time of the events within the city or other locale. Tapping on the stars displayed on city maps brings up the biographical page related to the specific locale, date, and event in Thomas Paine's life. The pertinent biographical information is presented in a scrollable/zoomable format, and illustrated with appropriate historical images.



Philadelphia, 1781-1783

With Cornwallis' surrender at the Battle of Yorktown on October 19th, 1781 serious hostilities with the British come to an end. On November 30th, living on Second Street in Philadelphia, Paine writes to Washington expressing disappointment over a lack of recognition, and confides to him thoughts of leaving America for France or Holland. Washington begins to make an effort to secure Paine some sort of compensation on his behalf. At the same time, Paine becomes friendly with Robert Morris through whom, in February 1782 along with Washington, Gouverneur Morris, and Robert Livingston, an agreement is made for Paine to be paid a salary of 800 dollars a year for writing in support of the war and efforts at creating a new constitution. Paine understands how the arrangement, which will only last until the end of the war, could be perceived as compromising his objectivity, but is comforted by the agreement that he will have final say over all his material. He uses the income to support new writings including *Crisis X*, which is in part a response to King George the Third's speech to Parliament in November of 1781 urging the continuation of the war despite the defeat at Yorktown. The work also argues for a stronger federal system, and a separation of federal tax and state tax, with Paine believing that "It has been our error, as well as our misfortune, to blend the affairs of each state, especially in money matters, with those of

Although a full peace would not be secured until the Treaty of Paris in 1783, the Battle of Yorktown ended hopes of the British of reestablishing authority in America.

Fig.4 Third ("City") level of the Biography navigation system with a biographical page open

Closing the particular biographical page automatically displays the next "event star" on the city map. When the events related to the particular city and time period have been exhausted, the "back" button in the bottom right of the city map blinks to remind the user to move to the upper level of the navigation interface in order to continue. Similarly, when all events related to a particular country/time-period have been exhausted, the user is prompted to return to the "world" map at the top navigation layer.

The biographical information can be traversed either chronologically or locationally. When using the default chronological traversal, the user is guided through the events in Paine's life one event at a time. The user has to navigate the multilayer map interface to locate the next event. If, on the other hand, the user is interested in locale-based information (e.g. which events in Paine's life transpired while he was in London), the "Clear Chronology" option displays all events pertaining to a particular locale without reference to a specific time period. The "Reset Chronology" option restores the default chronological traversal.

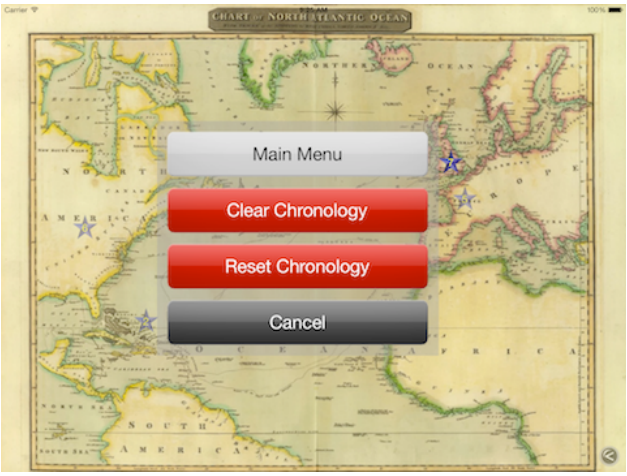


Fig.5 Biography component options

C. Selected Works Component

Thomas Paine was a true "Renaissance-man": a philosopher, a political thinker and activist, a revolutionary, a writer, an inventor, and even an engineer.[2] Of all his numerous accomplishments, the ones that his legacy is most firmly cemented on are his literary works. They ignited the American Revolution, supported ideas of the French Revolution, and continue to inspire people around the world to this day. Among Paine's most famous works are "Common Sense", "The American Crisis", "Rights of Man", "Age of Reason", and "Agrarian Justice". In addition to these, a large collection of over 350 of Paine's essays, letters, and other major works is stewarded by TPNHA, and is available through their website [13].

The limited hardware resources of a typical mobile device preclude the possibility of effectively storing the entire collection of Paine's writings – in proper format - as part of the TPE app. It was, therefore, decided to include in the application only the most significant and famous works of Paine, and to provide access to the entire TPNHA collection through a web-based interface embedded directly in the TPE application. The texts are accessed using the familiar moving-panels interface, which mirrors the main application interface. Tapping on the centered panel displays the full, historically-accurate text of the selected work, formatted to maintain the overall immersive look-and-feel of the app.

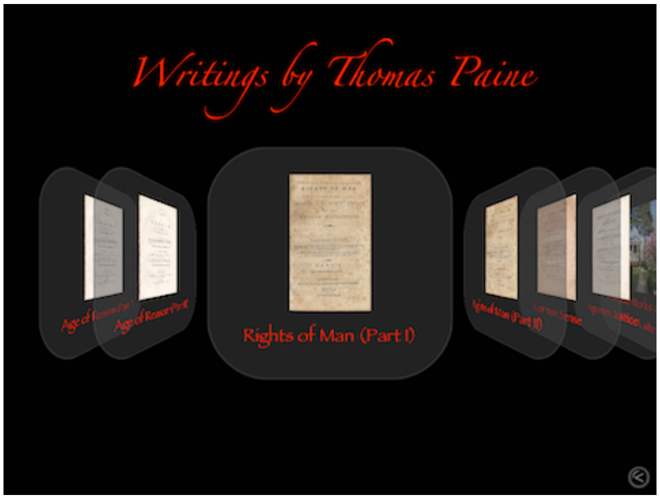


Fig.6 The Selected Works component interface

D. Anti-Paine Cartoons Component

Thomas Paine was an inherently controversial figure. Though admired by many, he was also deeply hated by rivals and political opponents, and considered a traitor by the British Government for supporting the American and French Revolutions. Paine's deist arguments against institutionalized religion and in favor of reason and free-thinking earned him enmity on both sides of the Atlantic Ocean, and especially in America.

[2] Paine designed bridges in England and in Philadelphia, US. He also worked on developing steam engines.

As a result, a large number of anti-Paine propaganda cartoons were created and published in the printed media of the time. TPNHA and other organizations and museums around the world archive the surviving engravings and paintings, and provide restricted access to them for educational and research purposes. The TPE application includes a relatively large sampling of the anti-Paine propaganda cartoons archived by TPNHA. The familiar moving-panels interface provides an overview of the available images.

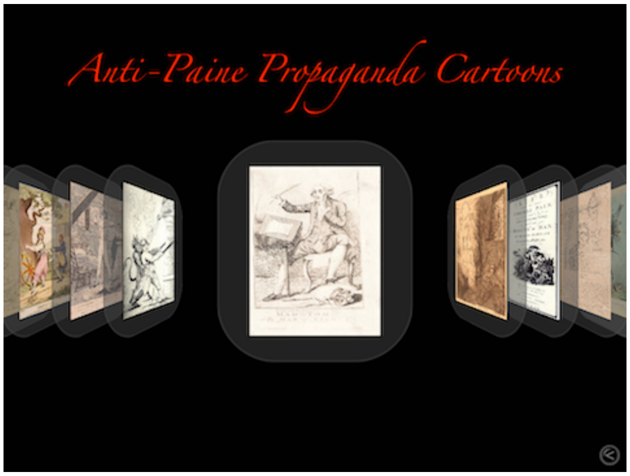
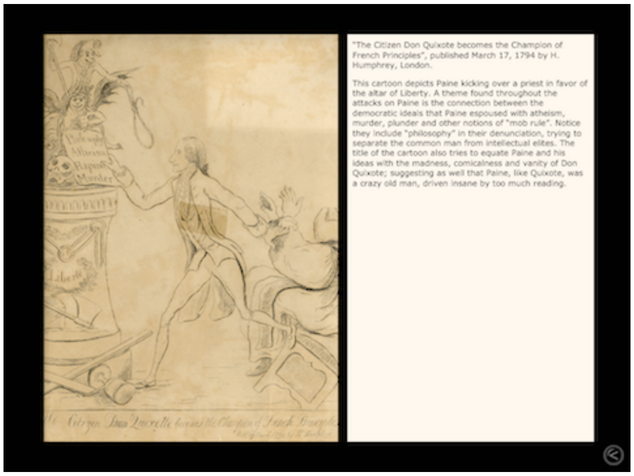


Fig.7 The Anti-Paine Cartoons component interface

As always, tapping on the centered image reveals the actual content – in this case a split panel with the selected image on the left and a write-up explaining the historical significance and circumstances of the image on the right.



"The Citizen Don Quixote becomes the Champion of French Principles", published March 17, 1794 by H. Humphrey, London.

This cartoon depicts Paine kicking over a priest in favor of the altar of Liberty. A theme found throughout the attacks on Paine is the connection between the democratic ideals that Paine espoused with atheism, murder, plunder and other notions of "mob rule". Notice they include "philosophy" in their denunciation, trying to separate the common man from intellectual elites. The title of the cartoon also tries to equate Paine and his ideas with the madness, comicalness and vanity of Don Quixote; suggesting as well that Paine, like Quixote, was a crazy old man, driven insane by too much reading.

Fig.8 A sample Anti-Paine Cartoons content page

E. *The Quotes Component*

Thomas Paine is renowned both for his witticism and for his profound, deep statements. Many of these statements have become widely popular, and are often used by literary researchers, authors, journalists, politicians, and common people alike. Gary Berton compiled an extensive set of Paine quotes for inclusion in the TPE application.

The quotes are grouped into a number of categories:

- Right to Revolution
- The American Revolution
- The French Revolution
- Citizenship
- War
- America
- Equality and Redistribution of Wealth
- Government and Rights
- Religion and Philosophy
- Monarchy
- Reason

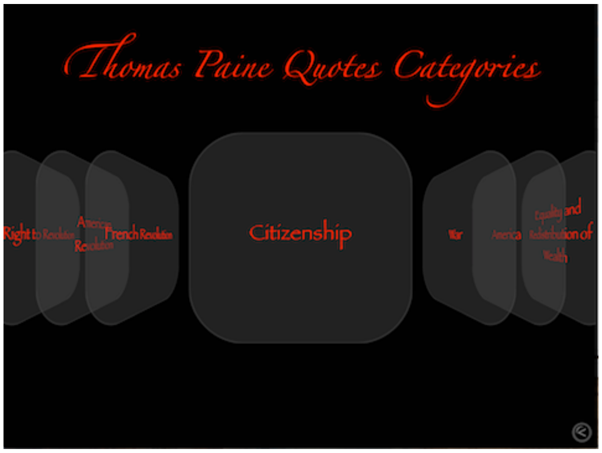
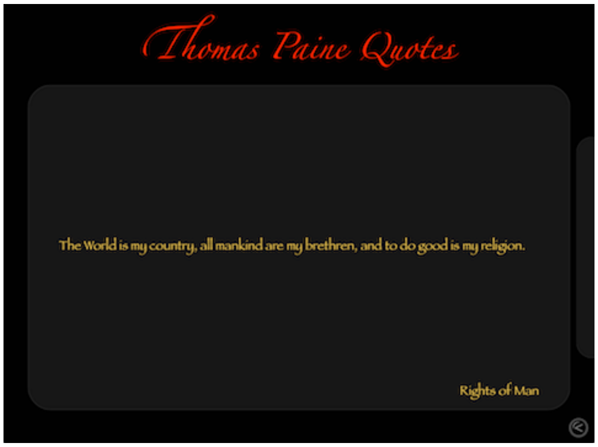


Fig.9 Paine Quotes component interface

Each category contains dozens of the most famous quotes by Thomas Paine. This categorization of quotes is unique to the best of our knowledge, and is extremely valuable for research purposes since it provides a direct access to the most topic-relevant quotations, avoiding the need for an exhaustive search through the entire collection.

The quotes in each category are presented using yet another variant of the familiar moving-panels interface. However, each panel is displayed full screen, and contains a single quote inscribed in golden letters. Swiping left or right slides the next/previous quote onto the screen using a modified version of the physics-based navigation system.



Thomas Paine Quotes

The World is my country, all mankind are my brethren, and to do good is my religion.

Rights of Man

Fig.10 A sample Paine quotation

V. ACADEMIC IMPACT

The application was deployed for free download on Apple's app store and has so far generated over 400 downloads. Understandably, the largest number of downloads comes from the US, Canada, and Europe, but a significant number of downloads originate in numerous other countries around the world - China, Australia, the Philippines, Indonesia, Japan, Cambodia, Nigeria, Ghana, India, Egypt, to name but a few. This in itself is a testament not only to the success of our project and the usefulness of the TPE application, but to the undying legacy of Thomas Paine whose ideas continue to inspire and move people in all corners of the globe.

The application was also well received in academic circles. Researchers studying the life and works of Thomas Paine have commented positively on the usefulness, completeness, and accuracy of the application. ITPS and Iona have recently introduced a new academic minor in Thomas Paine studies at Iona College, NY, and the TPE application will undoubtedly be a useful tool in many courses within the minor.

We are working hard to reach an even wider audience at universities and high schools, and to encourage the use of the TPE application in classes on American/World History, Political Science, Philosophy, and American Literature.

VI. CONCLUSION

In this paper we discussed the development of a unique and innovative mobile application, "The Thomas Paine Encyclopedia", devoted to the life and works of Thomas Paine. The application provides students and researchers with a complete collection of historically accurate materials presented in a visually attractive and easy to navigate format.

Directions for future work on the TPE application include:

- Providing access to the Paine works from within the biographical map interface, cross-referencing the literary materials with the location and time period of its creation on the appropriate historical map.
- Providing an expanded set of tools within the app, specifically intended for classroom use.
- Adding to the application a detailed and historically accurate section about the American Revolution, and cross-referencing it with the events in Paine's life.

A long term goal of our team is the development of similar applications devoted to other leaders of the American Revolution, and integrating the various applications into a educational and research suite available to scholars and students around the world.

ACKNOWLEDGMENT

We would like to acknowledge the essential collaboration of the following individuals closely involved in the development of "The Thomas Paine Encyclopedia" app:

- Gary Berton, ITPS/TPNHA – biographical text and image editing, historical consulting
- Stephen Pascuzzi, ITPS – biographical text and image editing
- Scott Cleary, ITPS – text editing and 18th century American literature consulting

We would also like to acknowledge the kind assistance of the following organizations, who contributed maps and images to "The Thomas Paine Encyclopedia" project:

- Cartography Associates LLC (David Rumsey Historical Maps Collection)
- Library of Congress
- US Army Center of Military History
- Harvard Map Collection
- American Philosophical Society
- TPNHA Collection at Iona College
- J. & C. Walker for Samuel Lewis's Topographical Dictionary, Robert Creighton
- Sussex Archaeological Society
- Educational Technology Clearinghouse
- Geographicus Rare Antique Maps
- Wikipedia Commons
- Margate Local History

Finally, we would like to thank Iona College, the Thomas Paine National Historic Association, and the Institute of Thomas Paine Studies for their continued support of this project and its future evolution.

REFERENCES

[1] S. Petrovic, L. Ivanov, G. Berton, R. Schiaffino, "Authorship Attribution of Thomas Paine Works", Presentation, Interdisciplinary Social Sciences Conference, Prague, Czech Republic, 7/13

[2] Aldridge, A. Owen, *Thomas Paine's American Ideology,* U. of Delaware Press, Newark, 1984

[3] Aldridge, A. Owen, *Man of Reason,* Cresset Press, London, 1960

[4] Ayer, A.J., *Thomas Paine*, Atheneum, New York, 1988

[5] Claeys, Gregory, *Thomas Paine: Social and Political Thought*, Unwin Hyman, Boston, 1989

[6] Conway, Moncure Daniel, *The Life of Thomas Paine,* 2 vols., G.P. Putnam's Sons, New York, 1893

[7] Edwards, Samuel, *Rebel!*, Praeger Publ., New York, 1974

[8] Foner, Philip, *The Complete Writings of Thomas Paine,* 2 vols., Citadel Press, New York, 1969

[9] Fruchtman, Jr., Jack, *Thomas Paine: Apostle of Freedom*, Four Walls Eight Windows, New York, 1994

[10] Hawke, David Freeman, *Paine*, WW Norton, New York, 1974

[11] Hawthorne, Hildegarde, *His Country was the World,* Longmans, Greene & Co., New York, 1949

[12] Hitchens, Christopher, *Thomas Paine's Rights of Man,* Atlantic Monthly Press, New York, 2006

[13] Online resource: http://www.thomaspaine.org/

A Study of Kinesthetic Learning Activities Effectiveness in Teaching Computer Algorithms

Sara Alraddady, Danny Luong, and G. Young
Computer Science Department, California State Polytechnic University, Pomona, CA 91768

Abstract

Kinesthetic learning is a teaching method that involves students' physical interaction among each other and the environment. The typical method of teaching, a classroom with an instructor talking and students listening and taking notes, has been the same for centuries. This research attempts to show that the kinesthetic learning activities (KLA) approach can be a viable alternative. In this study, the performance of students from an undergraduate level computer science course, Design and Analysis of Algorithms, is considered. In the winter 2014 quarter, selected topics were taught traditionally. The following spring quarter, these same topics were taught using KLA approach. The students' gained knowledge was measured in both quarters through pre / post tests. We hypothesized the KLA approach to be as efficient as the traditional lectures if not more efficient. The data collected proves our hypothesis to be correct. Our surveys also show that students enjoy the KLA approach.

Keywords: *kinesthetic, learning styles, teaching, algorithms.*

1. Introduction

Nowadays, technology is everywhere and in the hands of everyone. Although the benefits of technology in the learning process are immeasurable, technology itself has many drawbacks. One of these drawbacks is the increase in distraction among students. In most classroom settings, students just sit and listen to an instructor. It is easy for them to lose focus by checking social media or chatting with their friends. In addition, several studies suggest that students' attention during a lecture may last only up to fifteen minutes [8]. Therefore, it is challenging for some students to maintain concentration in a standard lecture.

Furthermore, people have different learning styles. According to Neil Fleming's model, there are four types of learners: The first group is visual learners who prefer learning through symbolic representations, using graphs and charts to obtain knowledge. Aural learners are those who perceive more knowledge when they are listening to a lecturer, and they have no problems learning in traditional lecture-based environments. There are also learners who prefer learning through text, either reading a book or writing notes. The last type is the kinesthetic learners who gain more knowledge through physical simulations for the concepts they are learning [3].

Active learning or kinesthetic learning approach is a solution for those who have trouble paying attention for a long period of time because it requires students to be physically active during the lecture. Since computer science topics are theoretical and intangible, grasping the concepts may be challenging for the students. Therefore, kinesthetic learning activities can be helpful and efficient.

1.1. Active learning characteristics:

Active learners tend to learn better through physical movement. Hence, they prefer performing arts and athletics more than studying theoretical science. They get distracted easily in a traditional lecture environment. They perfectly apply to the Chinese proverb which says "*I hear and I forget. I see and I remember. I do and I understand.*" These learners gain more knowledge when they use their hands and bodies. The KLA teaching method provides a suitable learning atmosphere because it forces learners to pay full attention in their movements. For example, a role playing KLA allows learners to imagine a situation where a problem arises, and they act like the solving agent. Another example is playing a game in the classroom like playing the tower of Hanoi puzzle. Kinesthetic learning gives an exciting learning experience for all learners, especially active learners.

1.2. Active learning techniques:

Faust and Paulson, from California State University, Los Angeles [2], have categorized active learning techniques into six types: First, exercises for individual students, which include quizzes that require the students to be in the classroom and pay attention to the lecture. Second, questions and answers, which is a method that allows the students to ask questions anonymously using the fish bowl technique, and the rest of the students answer these questions. This method empowers the students' role in their own learning process. Third, immediate feedback, which is exactly as its name suggests; it gives the professor an opportunity to measure the class's understanding as whole, and it is a helpful method that can be used in large classes. Fourth, critical thinking motivators; is a method that aims to get the students to think about the concept before they actually learn it using puzzles. The tower of Hanoi puzzle is a good example of this technique. The concept can be explained to students before they attempt to solve the puzzle. The learning process actually happens when they solve the puzzle with the algorithm that they have been provided. The fifth active learning type is share/pair. This method forces students to work together as pairs by exchanging their thoughts and opinions and explaining unclear concepts. An example of this learning type is pair programming which always produces high quality software [9]. The last technique is the cooperative-learning strategy, which requires students to be divided into groups. Each group is responsible of solving a given problem and explaining it to the rest of the class. This method enhances students' communication and social skills.

1.3. Benefits:

The typical classroom setting, where students are just passively sitting and listening, leads students to enter their "relaxed zone," where it is much harder for them to maintain full attention. On the other hand, a kinesthetic activity fills the room with energy and excitement, and it makes students see things from a new angle. Moreover, a KLA also helps students to develop interpersonal skills, since it requires them to communicate with each other. It helps timid students to interact with their classmates in an observed environment. As reported by the National Training laboratories' pyramid of learning, students gain only 5% of the information given to them in the form of lectures while learn-by-doing retains 75% of the knowledge. Furthermore, students retain 90% of what they have learned when they teach each other [6].

1.4. Obstacles:

Despite the numerous advantages of active learning, some instructors are reluctant to adapt this teaching method for several reasons. For example, how an instructor decides to manage their time has a major influence of the whole learning process and its outcome. An instructor must account for the preparation and execution time, as well as students' responses to such new learning strategy. Some instructors think that they do not have enough time to cover all the topics they have assigned for a certain quarter, and active learning can reduce the amount of available time. As a result, an instructor may conclude that lecturing is more convenient for delivering the information. However, a scenario where an instructor explains and students listen does not guarantee that students will be able to absorb the knowledge. Students may leave their classroom with some bits and pieces of a lecture in their notebooks and with nothing in their heads. Also, there is a big chance that a number of students may stop the instructor to ask question, and the instructor might end up not covering every topic that needs to be covered. In addition, in a traditional lecture, the teacher has more control of the class where in an active learning session; students are more involved with how the course is run. With this in mind, an instructor may be ambivalent to use KLA as new problems with classroom management may arise.

2. Past Research:

In the past decade, kinesthetic learning activities were common in preschools and elementary schools. In 2004, Tammy Nguyen did a study named "Do kinesthetic strategies influence students' achievement?" She studied the effectiveness of teaching mathematics kinesthetically for first-graders. She taught them addition and subtraction kinesthetically using hand signals and traditionally, and she tested them. The results showed that the kinesthetic test scores were higher than the non-kinesthetic test scores. Also, she noticed that the students were excited about participating in the KLA's. Although the significant findings of the benefits of kinesthetic learning at the preschool and elementary level, this study by itself can-not prove that kinesthetic learning is more effective than traditional lecturing [5].

In 2009, Katherine Gunion taught middle school students the concept of recursion in a KLA fashion. She held a 7-weeks-after-school program for the students and used 6 different KLAs. She wanted to answer three questions:

1- Can students identify recursion?

2- Can the students understand and apply recursion?

3- What is the effect of understanding recursion on their attitudes towards computer science?

Her answers were as follows: most of the students were able to identify recursion, but few of them were able to apply it on solving different problems instead of solving them sequentially. For the last question, the different data collecting methods she used showed that the students enjoyed the activities, and half of them came back for the next offering of the program, but this still does not prove our goal [4].

Similarly, for college students, there were several publications about kinesthetic learning activities in the computer science field. "Human cons cell jeopardy" which is a KLA developed by Begel, Garcia, and Wolfman to introduce a lisp programming course [1]. Moreover, in 2007, Sivilotti and Pike developed a set of KLAs for both undergraduate and graduate students to teach them the concepts of distributed systems [7]. In general, there are only a few KLAs that have been designed for computer science education at the university level.

3. Research Goal and Methodology:

In this study, we applied kinesthetic learning strategies on students enrolled in a Design and Analysis of Algorithms course in the Computer Science department at Cal Poly Pomona University to test the efficiency of the KLAs compared to traditional lectures. Our hypothesis says that the kinesthetic learning will be as effective as traditional lecturing, if not more effective. It is important to mention that this research has been revised and approved by the Cal Poly Pomona Institutional Review Board (IRB): protocol #14-0004, and it has met the federal and state regulations and Cal Poly Pomona policies regarding the safety of the human subjects who participated in this research.

In winter 2014, the students in Dr. Young's class CS331 Design and Analysis of Algorithms participated in this study.

The chosen topics for this study were Human Sorting, Median of the Medians Divide-and-Conquer approach, and 0/1 knapsack problem Dynamic Programming solution. The students were lectured traditionally, and their background knowledge about the topics was measured using a pretest. Also, they were given a posttest after the lecture to see how much information they gained. During the following quarter spring 2014, students enrolled in the same class were taught the same topics through KLA. They were given pretests and posttests as well. After completing the posttests, the students surveyed their thoughts and opinions about the teaching method. The pretest results along with the posttest results were analyzed and compared to determine the performance of both traditional lecturing and KLA.

3.1. Human Sorting:

There are four distinguished types of sorting algorithms: bubble sort, selection sort, merge sort, and quick sort. Twenty-five students participated in this activity. The students were divided into four groups (six to seven students per group). Each group was responsible for learning the assigned algorithm using the textbook or the Internet. They were given approximately 20 minutes. After that, they were asked to sort themselves kinesthetically according to their heights in an ascending order using the assigned algorithm's strategy in front of the class. After each demonstration, an explanation was presented by the instructor to make sure the rest of students understood the strategy and the complexity of the algorithm.

3.2. Median of the Medians – Divide and Conquer Approach:

This exercise solves the following problem: Given a list of n elements, find the kth smallest element with worst case time complexity *O(n)*. Twenty one students were involved in this KLA. Six students acted as judges for the students' height. The rest of the students formed a table of three rows and five columns. Each column represented a group. Each group was asked to sort themselves in an ascending order with the help of the judges with the tallest student at the back seat. After that, each student in the middle row was announced as the group leader. Next, the leaders were asked to sort themselves in front of the class beginning with the tallest student on the left (facing the rest of the class). Once the leaders were sorted, they were asked to sit in the new column they belonged to. The rest of the seated students were asked to follow their leaders with the original order of the column. The student who was sitting in the middle column and the middle row was declared as the median of the medians. When facing the students, the bottom right quarter of the students was shorter than the median of the medians or at most as tall as him /her, and the top left quarter was at least as tall as the median of the medians as illustrated in figure 3.2.

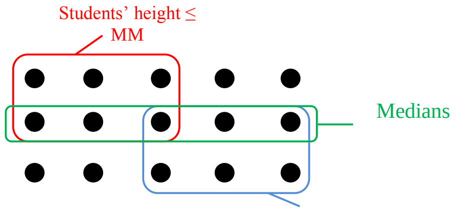


Figure 3.2.

3.3. 0/1 knapsack problem – Dynamic Programming Solution:

This particular activity solves the following problem with dynamic programming approach: "Giving a set of *n* items each with weight *w* and profit *v*, determine the items with most possible total profit to include in a set without exceeding the set's weight limit *W*." The data for this exercise is as follows: *n* = 3, *W* = 5. According to the problem's data, we needed to create a four by six table, where each student represents a cell in the table. For this activity, the more students are involved, the larger the table will be needed. Nineteen students participated in this exercise. Fifteen students were asked to form a human table, and each one of them was given a cup labeled with its capacity. Because the first row represents having zero items, the students were asked to leave the first row empty, and the same was asked for the first column because it illustrates having a knapsack with zero capacity, so no item will fit in this capacity. Three other students were asked to act like the objects, and the last remaining student was asked to flag the cells that chose to keep the object. After forming the table and explaining what the rows and columns represent, each student representing an object was asked to go to each column in his/her row and ask them if the object fits in their cell's capacity and whether they chose to keep it or not. If the object fit and they chose to keep it, the student representing the object placed the object into the cup. The object holders students needed to go check with their columns one after another. This means that object two had to wait until object one finished checking with each column in his row before start checking his/her row cells. Once a cell chose to keep the object, the flag student marked the cell as chosen. This step was important to trace back the optimal solution. Besides explaining to the students what each row and column represent, the students were given a scenario sheet to help them decide whether to keep the object or not. After filling the table, the students traced back the flagged cells by checking the leftover capacity to choose the optimal solution.

4. Research Findings:

With the contributions of about 60 students in two quarters, we were able to compare the effectiveness of KLAs to the effectiveness of the traditional lecturing. The results showed that KLAs are more effective, and the students gained more

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

knowledge. Furthermore, the survey, which was given to the students in spring 2014, demonstrated that the students enjoyed the KLA, and their comments were positive about this type of learning.

The middle 50% of students' pretest scores in the traditional lecture section is shown in table 4.1. Students #8 to #20 scored an average of 56.41% (6.77 out of 12) while the middle 50% posttest scores from students #7 to #17 scored an average of 85.6% (10.27 out of 12). By comparing the pretest and posttest scores, we can see that the learning gained through the traditional lecture increased by 29.19%. On the other hand, the middle 50% of students' pretest scores in the KLA section in table 4.1. from student #8 to #21 averaged 55.83% (6.7 out of 12), and the middle 50% posttests scores from student #7 to #17 averaged 90.3% (10.90 out of 12). By comparing the results of the tests, we can see that the students gained knowledge increased by 34.47% through KLA.

Respectively, by taking the ratio of the gained knowledge from both quarters, KLA was 1.2 times more effective than the traditional lecture method for the 50% middle students. The pretest scores were higher because of a previous disclosure of the algorithms in prerequisite courses.

From the scores shown in table 4.2, the middle 50% of Median of the Medians traditional lecture' pretests, which ranges from student #7 to #17, scored an average of 12% (0.36 out of 3), and the middle 50% for the posttest of the students' range from #7 to #17 scored an average of 21%. In contrast, the middle 50% of the pretest scores for the students from #8 to #20 in the KLA section scored an average of 30.7% (.92 out of 3). Furthermore, the middle 50% posttest scores of the students from #8 to #20 scored an average of 54% (1.62 out of 3). As a result, the students gained 23.3% knowledge from the KLA and 9% from the traditional lecture. Comparing the gained knowledge from both classes, there was a 14.3% increase through KLA.

As a result, after taking the ratio from both quarters for the two learning gains, KLA was 2.6 times more effective than the traditional lecture for the 50% middle students. Furthermore, most of the students were able to find the median of the medians in the pre/posttests, and only a few were not able to calculate the suitable number of the required subsets.

Lastly, the middle 50% pretests scores for the students ranged from #6 to #14 in the traditional lecture section scored an average of 1.38% (0.11 out of 8), and the average of the middle 50% of the students posttest scores from student #7 to #15 was 59.75% (4.78 out of 8). Thus, the students overall gained knowledge was about 59.25%. Moreover, the 50% middle pretest scores in the KLA section, which ranges from student #7 to student #17 averaged 0% (0 out 8), and the middle 50% of the posttest scores for the range of students from #8 to #20 scored an average of 67.37% (5.39 out 8). Accordingly, the middle 50% showed a big improvement, and their overall gained knowledge scored up to 67.37%.

	Traditional lecture		KLA	
Student	Pretest	Posttest	Pretest	Posttest
1	0	3	0	5
2	1	4	1	9
3	3	7	2	9
4	3	7	2	10
5	3	8	2	10
6	4	9	3	10
7	5	9	3	10
8	5	9	5	10
9	6	9	5	10
10	6	10	5	11
11	6	10	5	11
12	6	11	6	11
13	6	11	7	11
14	7	11	7	11
15	7	11	7	11
16	7	11	7	12
17	8	11	8	12
18	8	11	8	12
19	8	12	8	12
20	8	12	8	12
21	9	12	8	12
22	10	12	8	12
23	11	-	10	-
24	11	-	10	-
25	12	-	10	-
26	12	-	11	-
27	-	-	11	-
28	-	-	11	-
Average	6.61 51.3%	9.55 79.6%	6.35 53.3%	10.59 89%
Max	12 100%	12 100%	11 91.6%	12 100%
Min	0 0%	3 25%	0 0%	5 41.6%

Table 4.1 Tests Results for Sorting Algorithms

Consequently, after taking the ratio from both quarters for the two learning gains, KLA was 1.2 times more effective than the traditional lecture for the 50% middle students. The reason for such a high result is because all of the students contributed to the activity, as well as due to the repetitive nature of the exercise.

	Traditional Lecture		KLA	
Student	Pretest	Posttest	Pretest	Posttest
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	1
7	0	0	0	1
8	0	0	0	1
9	0	0	1	1
10	0	0	1	1
11	0	1	1	1
12	0	1	1	1
13	0	1	1	2
14	1	1	1	2
15	1	1	1	2
16	1	1	1	2
17	1	1	1	2
18	1	1	1	2
19	1	1	1	2
20	1	1	1	2
21	2	1	1	2
22	3	2	1	2
23	-	3	1	2
24	-	-	1	2
25	-	-	1	3
26	-	-	2	3
27	-	-	-	3
Average	0.54 18%	0.69 23%	0.73 24.3%	1.5 49.4%
Min	0 0%	0 0%	0 0%	0 0%
Max	3 100%	3 100%	2 66.6%	3 100%

Table 4.2 Tests Results for Median of Medians

	Traditional lecture		KLA	
Student	Pretest	Posttest	Pretest	Posttest
1	0	0	0	1
2	0	0	0	1
3	0	0	0	1
4	0	0	0	1
5	0	1	0	2
6	0	1	0	2
7	0	2	0	2
8	0	3	0	3
9	0	3	0	3
10	0	5	0	3
11	0	6	0	4
12	0	6	0	4
13	0	6	0	5
14	1	6	0	6
15	1	6	0	7
16	1	7	0	7
17	2	7	0	7
18	2	7	0	7
19	-	8	0	7
20	-	-	0	7
21	-	-	0	7
22	-	-	1	7
23	-	-	-	8
24	-	-	-	8
25	-	-	-	8
26	-	-	-	8
Average	0.38 4.75%	3.89 48.6%	0.04 0.5%	4.84 60.5%
Min	0 0%	0 0%	0 0%	1 12.5%
Max	2 25%	8 100%	1 12.5%	8 100%

Table 4.3 0/1 Knapsack problem tests results

Table 4.4 demonstrates the learning gains for both teaching methods for the counted 50% middle students, and the ratios showing the effectiveness of KLA in comparison to the traditional lectures.

Method	Sorting Algorithms	Median of the Medians	0/1 Knapsack
Lecture	21.78 %	10.3%	54.12%
KLA	34.47%	22.3%	70.25%
Ratio:	1.2	2.6	1.2

Table 4.1 Ratios of the learning gains lecture vs. KLA for the 50% middle students

When we focus on the whole class's performance, Figures 4.1 - 4.3 illustrate the improvements of the pretest' and posttest' averages for both quarters. We expected the pretests averages to be low because they were given to the students to have an idea what they already know about the concepts they will learn before they learn it. However, some topics were not entirely new for some students as the case of the sorting algorithms in Figure 4.1.

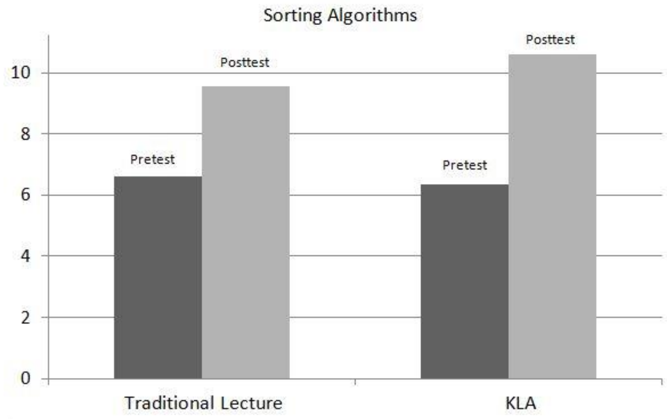


Figure 4.1 Sorting Algorithms average tests results

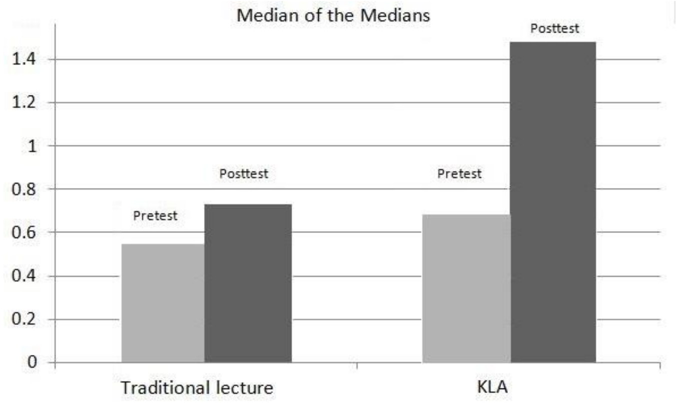


Figure 4.2 Median of the medians average tests results

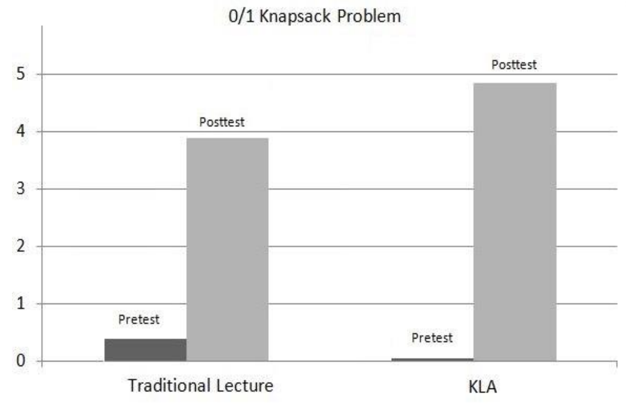


Figure 4.3 0/1 Knapsack problem tests average results

The posttests were given after the topics were taught to gauge the students' understating of the topic. Comparing the three figures, the sorting algorithms posttest averages among the two quarters were the highest. Once more, this was due to the students' previous exposure to the topic.

5. Survey Feedback:

In spring 2014, a total of 30 students reviewed the KLA with a one-page survey that consisted of short answers and liking scales. According to table 5.1, 46.6% of the surveyors chose kinesthetic learning as their preferred learning style, and 68% of them evaluated the KLA as a good teaching method. The last two questions in survey asked the students to write their opinion on what they liked and what they did not like about the KLA. One student wrote: "Just like all teaching techniques it is subjective." Usually, what makes a difference to a student's ability to learn is how the concept was taught, not what was being taught. Another student wrote regarding KLA: "Hard to lose focus during such activities." Because KLA depends on students' physical movements, students will not have a chance to lose focus or fall asleep. Another student said: "KLA is more interesting and makes things simpler to learn, but it can be rather time consuming." A well designed KLA and a prepared instructor are two factors that reduce the time wasted that KLA may cause.

Learning Style	Percentage
Visual	33.3 %
Auditory	3.3 %
Writing	0 %
Kinesthetic	46.6 %
I don't know	16.6%
Total	30

Table 5.1 Surveyors Learning Styles

6. Conclusion and Future Work:

Kinesthetic learning activities are more effective than the traditional lectures according to table 4.1. This result is similar to the results of Nguyen's study, Gunion, Begel, Garcia, and Wolfman. Moreover, the use of KLA is a beneficial tool for helping students to understand the elusive concepts of computer science. Furthermore, KLA can be a good solution for helping students to stay focused and not get distracted during the class. When utilyzed for a well designed activity and to focusing on the topic at hand, KLA can be a powerful tool to teach theoretical concepts in computer science.

Likewise, related studies can be done to determine the effects of a particular form of KLA activity and its effect' on the students' understanding of a topic. Also, applying this teaching method on different subjects can help support the results of this research.

REFERENCES

[1] Begel, Andrew, Daniel D. Garcia, and Steven A. Wolfman. 2004. "Kinesthetic Learning in the Classroom." *35thProceedings of the Technical Symposium on Computer Science Education*. New York, NY: 183-184.

[2] Faust, Jennifer L., and Donald R. Paulson. 1998. "Active Learning in the College Classroom." *Journal on Excellence in College Teaching*, 9(2): 3-24.

[3] Fleming, Neil D. 2001. "Teaching and Learning Styles: VARK Strategies." Christchurch, New Zealand: Author.

[4] Gunion, Katherine. 2009. "FUNdamentals of CS: Designing and Evaluating Computer Science Activities for Kids." Victoria, BC, Canada: MS thesis, University of Victoria.

[5] Nguyen, Tammy. 2004. "Do Kinesthetic Strategies Influence Students‟ Achievement?" Pomona, CA: MS thesis, California State Polytechnic University, Pomona.

[6] Polovina, Simon. 2011. "About the Learning Pyramid." http://homepages.gold.ac.uk/polovina/learnpyramid/about.htm (accessed December 1, 2013).

[7] Sivilotti, Paolo and Scott M. Pike. 2007. "The Suitability of Kinesthetic Learning Activities for Teaching Distributed Algorithms." *38th Proceedings of the Technical Symposium on Computer Science Education*. New York, NY: 362-366.

[8] Wankat, Phillip C. 2002. "The Effective, Efficient Professor: Teaching Scholarship and Service." Boston, MA: Allyn and Bacon.

[9] Williams, Laurie, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. 2000. "Strengthening the Case for Pair Programming." *IEEE Software*. 17(4): 19-25.

Critical Thinking Strategies in Teaching Advanced Timing Analysis Course

J. Pang[1], **A. Q. Liu**[2]
[1]Department of Electrical and Electronic Engineering and Program of Computer Engineering,
California State University, Sacramento, Sacramento, CA, U.S.A.
[2]Department of Sociology and Office of Academic Program Assessment,
California State University, Sacramento, Sacramento, CA, U.S.A.

Abstract - *Due to the increasing design complexity and time-to-market pressure for digital integrated circuit design, the efficient static timing analysis method based circuit analysis becomes essential in circuit design nowadays. In this paper, we propose using critical thinking strategies to teach advanced timing analysis to electrical engineering and computer engineering graduate students. Traditionally, graduate students only passively receive information inside classroom without developing critical thinking skills for problem solving. Critical thinking occurs when students formulate hypothesis, synthesis information from thorough analysis of problems, articulate their thoughts and finally draw logical conclusions. In addition, interaction with team members in class engages students in course materials, foster critical thinking, and enhance their learning process. This paper is to present the importance of critical thinking for graduate student education, and describe multiple critical thinking strategies to innovate teaching, to engage and motivate students for better learning.*

Keywords: Critical thinking, computer engineering education, electrical engineering education, static timing analysis, team based learning

1 Introduction

Traditionally, instructors play dominant roles in engineering education and students receive knowledge inside classroom passively. Most of the time, students acquire ideas through memorizing facts. During such passive learning process, students are not given enough time or they may not feel comfortable for raising questions, promoting class discussion, analyzing and evaluating information.

Passive learning is the result of instructor-centered one-way communication in teaching. Without getting verbal feedback from students, the instructors usually need much more time and greater effort to find out strength and weakness of students in studying course materials. Moreover, because students only absorb information without much effort for comprehension or understanding, such routinely memorizing activity does not sharp their brain for questioning, thinking and analyzing. Consequently, students do not acquire critical thinking skills and they are not prepared to solve complex problem in their future career.

In recent years, with an increasing market demand for graduate students to develop reasoning and critical thinking skills through academic studies [1], teaching methodologies in engineering education have trends of integrating lectures with active participation of students through critical thinking and collaborative learning [2][3][4][5].

Critical thinking changes instructor-centered learning into actively involving students in learning. To develop critical thinking skills, students are encouraged to improve attention and observation of the problems from multiple angles. They are motivated to enhance their understanding of information through their own questioning, analysis and evaluation process. In general, critical thinking has direct impact on effective and logical decision making for students. In addition, such practice allows students to come up with solutions to new scenarios and also complex problems in their future career. It facilitates life-long learning and personal growth.

Although surveys show that above 90% faculty agree that critical thinking is one of the primary goals of higher education, development of critical thinking skills is generally considered important and emphasized in undergraduate education [6]. More effort for critical thinking in the graduate study especially in the engineering field needs to be conducted and the results need to be distributed.

With the increasing design complexity and time-to-market pressure for digital integrated circuit design, using efficient static timing analysis method to verify and optimize the performance of the digital circuit is emerging as a critical approach for the success of the application-specific integrated circuit (ASIC) design [7][8][9]. In addition, the explosive growth of the digital circuit design and rapid technological changes require graduate students in the computer engineering field and the electrical engineering field to be able to analyze and evaluate information, and also think critically to solve complex problems in their future career.

In this paper, critical thinking strategies in teaching advanced timing analysis course to graduate students in Electrical Engineering and Computer Engineering at California State University, Sacramento (CSUS) are discussed in detail. Part 2 discusses how to provide design problems to stimulate critical thinking among students. Part 3 presents team based learning strategy to engage students on critical thinking. Part 4 covers the usage of computer tools for critical thinking. Part 5 describes the strategy of using writing assignment of technical paper review to motivate students for critical thinking. Finally, part 6 draws conclusions of this work.

2 Problem design for critical thinking

In order to change instructor-centered learning into active learning requiring critical thinking, graduate students should be provided with opportunity to reason course contents, conduct student-centered learning and derive logic conclusions by themselves. Proper design of questions for students to practice critical thinking is critical. The questions should arouse intellectual curiosity and also encourage deeper investigation from students. Meanwhile, the questions should be mapped into course learning outcomes.

In student-centered learning, proper guidance is still required from instructors to prepare students to enter critical thinking stage, to evaluate student study results and help students improve learning.

Before working on more complicated critical thinking assignment, students need to first participate in some warm-up activities. For example, the first exercise is for students to follow a tutorial developed by instructor to learn how to use Synopsys static timing analysis tool of Design Vision. The second exercise is for students to study one set of example design constraints provided by textbook [8] using Design Vision online help tool. Students are required to summarize the usage of the following clock design constraints in a table after completing the second exercise: *create_clock*, *set_clock_uncertainty*, *set_clock_latency*, *set_input_delay*, *set_output_delay, and set_clock_transition*.

After completing these warm-up exercises, graduate students are usually eager to explore deeper and move on to more challenging questions. One example of critical thinking related problem assignment given by the author in advanced timing analysis course is described below to further stimulate students on active learning.

In static timing analysis, a circuit is converted to a timing graph to represent timing paths. A timing path is defined as a path from a valid start point to a valid end point. The start point can be an input port, or a clock pin of the D Flip-flop. The end point can be an output port or a D pin of the D Flip-flop. All possible timing paths need to be checked against the design constraints. This assignment is to use the Synopsys Design Vision tool to check timing violations with design constraints developed by each group of students. Students are required to use Verilog Hardware Description Language to design the circuit in Figure 1 for static timing analysis based on the Synopsys tool. Students are also asked to form groups for discussion and each group should meet with instructor face to face to discuss final results. This assignment has following six learning outcomes.

1). Learn how to use Synopsys Design Vision tool;

2). Learn how to use timing constraints;

3). Develop a comprehensive analysis of a given circuit in Figure 1 to find critical timing path, setup and hold violations.

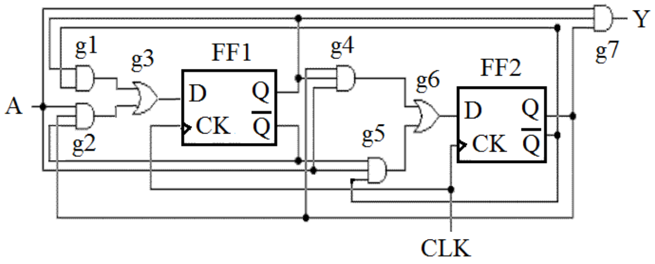


Figure 1. A sequential circuit with multiple gates

4).Thoroughly analyze the assumptions of timing constraints, and carefully evaluate the relevance among gate level circuit timing, setup condition and hold condition.

5).Formulate a specific hypothesis of circuit timing equations which accounts for the complexities of the setup and hold violation issues. Use Design Vision tool to verify the hypothesis. Four or five students form a group for discussion. Acknowledge the limits of the position and synthesize others' points of view in each group.

6). Draw logical conclusions and related outcomes of the assignment.

After graduate students finish their work, the feedback discussion with instructor allows students to confirm their ideas, to correct errors, and also to go back to work on the assignment again to fix errors. The feedback loop makes critical thinking work of students more conductive.

Instead of providing solutions as lecture materials directly to students in the traditional instructor-centered teaching, the instructor created this assignment to engage students in critical thinking, active learning and also achieve major student learning outcomes. By working on such assignment, students benefit from improving critical thinking skills to analyze and synthesize course materials with practice.

3 Team based learning

Developing critical thinking skills require time from graduate students to practice and also require courage from them to face complex problems. Usually graduate students with different academic background have different learning styles, some are slower learners and some are faster learners. To help graduate student retention in the classroom, to help graduate student better manage their time for study, and also to balance learning environment inside the classroom, team based learning is very useful.

In advanced timing analysis graduate course, usually four or five students form one team for practice. The instructor's primary role is to design and manage the overall instruction process. In addition, the instructor needs to monitor the study behavior of graduate student teams and provide feedback to student team work through face-to-face discussions. Each team is allowed to distribute work among students.

In advanced timing analysis graduate course, the instructor developed unique model of *tutorial-critical thinking practice-feedback loop* to organize team assignments and discussions. Tutorial exercises are usually designed by instructor to get each team of students familiar with general concepts related to course materials and also the critical thinking practice problems.

The tutorials are straight-forward exercises for students to follow. They are used to warm up students. Generally speaking, the completion of tutorial work introduces the sense of academic success in students, and also motivates them to try harder assignments for the next step.

The critical thinking practice problems are designed to emphasize student learning outcomes outlined in course syllabus. Each problem is of sufficient difficulty for a team of four or five students to work together, conduct experiments, collect data, evaluate results, derive reasonable conclusions and enable significant learning.

During the team work, graduate students learn from each other, argue with each other, evaluate views of others in the same team, solve problems through critical thinking and team discussions, and finally make logical decision to reach a consensus. Such practice is not only good for graduate students to cultivate critical thinking skill, but also improve their oral communication skill. Both are highly desirable employee characteristics in the future industry working environment.

Finally, each team is also required to articulate their work in a written report which allows students to document design work, review technical design details, analyze and evaluate the whole design process, make the thinking of team members visible, and also draw the final logical conclusions. The instructor needs to provide report feedback through face-to-face discussion. Such practice enhances critical thinking skills of students and also helps improve their writing skills.

4 Usage of computer tools

The usage of computer tools is extremely beneficial for promoting critical thinking among graduate students in the engineering field. Computer tools allow graduate students to model different scenarios for a problem and visualize tangible results. In addition, computer based visualization help students organize information in brain for useful future retrieval. Computer tools make study more interesting for graduate students. As a result, the usage of computer tools can bring strong sense of accomplishment to graduate students, stimulate them to participate in the critical thinking process, allow them to evaluate ideas more objectively, and speed up their problem-solving process.

The activities of using computer tools for teaching need to be well planned to ensure quality instruction, to engage students in course materials, and to facilitate student learning outcomes.

For example, the advanced timing analysis course at CSUS is majorly related to ASIC design. Students are expected to learn industry tools for static timing analysis. The usage of computer tools in this course is based on the simplified ASIC design flow chart shown in Figure 2.

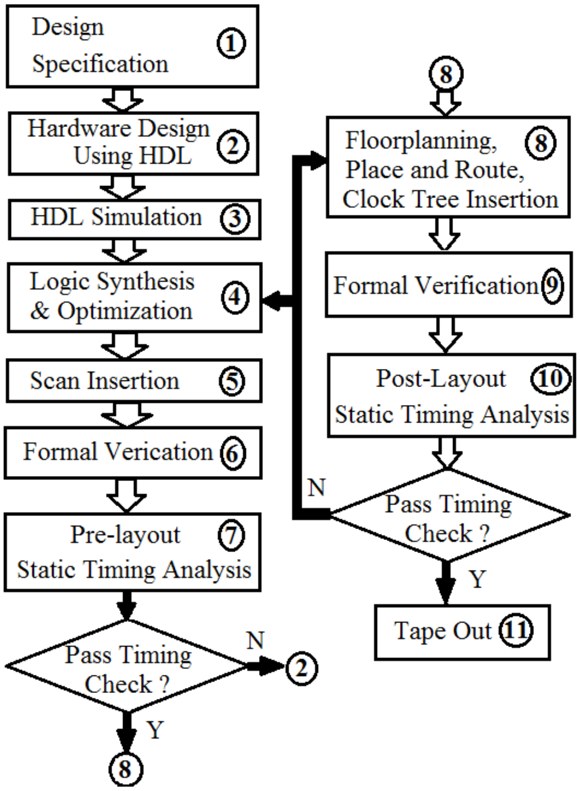

Design Specification ①
Hardware Design Using HDL ②
HDL Simulation ③
Logic Synthesis & Optimization ④
Scan Insertion ⑤
Formal Verication ⑥
Pre-layout Static Timing Analysis ⑦
Pass Timing Check ?
N
②
Y
⑧
⑧
Floorplanning, Place and Route, Clock Tree Insertion ⑧
Formal Verification ⑨
Post-Layout Static Timing Analysis ⑩
Pass Timing Check ?
N
Y
Tape Out ⑪

Figure 2. Simplified ASIC design flow chart

Since the course is about static timing analysis of the ASIC design, physical floor planning, place and route in Figure 2 are not covered by this course.

Typical assignments in this graduate course include following.

1). HDL simulation using Perl and Synopsys VCS simulator tools in step 2 of the above flow chart.

2). Pre-layout and post-layout static timing analysis assignments based on step 7 and step 10 of the above flow chart using Synopsys Design Vision and PrimeTime tools. Students also need to learn how to use TCL tool to write design constraints for static timing analysis.

All of above computer tools are widely used by industry for ASIC design. Consequently, most graduate students feel strongly motivated for study and stimulated to apply critical thinking for design assignments using computer tools.

5 Writing technical paper review

Engineer learning is a life-long process due to the fast development of technology. Although reading technical papers is quite challenging, it is essential for graduate students to develop critical thinking and problem-solving skills in their future career and also their life-long learning.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

In advanced timing analysis graduate course, each individual student is required to write a technical paper review regarding digital circuit static timing issues. One IEEE paper titled “Type-Matching Clock Tree for Zero Skew Clock Gating”[10] provided by the instructor is available for graduate students to review. Moreover, each graduate student is also allowed to choose a different technical paper related to the static timing analysis issues based on personal interest.

The writing instruction provided by the instructor is designed to ensure the review writing to serve as a product communicating the result of critical thinking. It requires graduate students to include five parts shown below in their technical paper review.

Part 1 of the technical review writing is instruction part. In this part, graduate students need to clearly identify the major findings of the technical article including major methodology, major concerns, major data collection, major results and so on.

Part 2 is for students to clarify the hypothesis. In this part, graduate students summarize the problem and the proposed solution employed in the technical paper.

Part 3 requires students to collect engineering data by presenting the main figures, schematics, tables, logic diagrams, truth tables, Boolean algebra, logic equations, waveform and state diagram used in the technical paper.

Part 4 asks students to explain the functionality of the engineering data, analyze whether the methods used in the technical paper can apply the engineering data to achieve the design purpose or solve particular problems. Moreover, students are required to evaluate if the technical paper makes any interesting points or controversial points, and also evaluate if the technical paper makes major contributions to new ideas, new experimental methods, new survey and so on.

In Part 5, students need to draw a conclusion about what they have learnt from their reading.

The above technical paper review assignments require graduate students to evaluate other people’s view, provide reasoning to support their thought, and make their ideas explicit. Such exercises motivate graduate students, stimulate their interests in the subject and improve their critical thinking skills. The technical paper review is no longer team based learning, instead it is individual work.

6 Conclusions

Critical thinking strategies in teaching advanced timing analysis course to graduate students in electrical engineering and computer engineering at CSUS are proposed in this paper.

First of all, the proper design of problems to encourage critical thinking among graduate students allows transfer from instructor-centered teaching to student-centered active learning.

Second, team based learning strategy helps organizing critical thinking activities and facilitates achieving conductive student outcomes. This paper presents one unique *tutorial-critical thinking practice-feedback loop* model applied to team based learning in the engineering education field for critical thinking.

In addition, this paper discusses the usage of computer tools for critical thinking based on course objectives, student learning outcomes, and industry application trend.

Finally, this paper presents how to use writing assignment for technical paper review to enhance critical thinking skill in the engineering field.

In conclusion, critical thinking strategies help engage students in active learning inside classroom, and also prepare students for complex problem solving in their future career as well as in their life-long learning.

7 References

[1] Goh Wei Wei. “The Use of Wiki to Facilitate Critical Thinking”; 2012 IEEE International Conference on Teaching, Assessment and Learning for Engineering, Pages H3C-1 – H3C-3, Aug. 2012.

[2] M. G. Rasteiro. “Interactive Simulators: A Contribution to Link Theory and Applications in the Teaching of Chemical Processes”; 2013 1st International Conference of the Portuguese Society for Engineering Education, Pages 1 – 8, Oct. 31st – 01 Nov. 1st 2013.

[3] Chien-Hung Lai, Bin-Shyan Jong, Yen-Teh Hsia, Pai Feng Lee, Te-Yi Chan. “Systematic Analysis of Peer Interaction in Cooperative Learning”; 2011 Frontiers in Education Conference, Pages S4E-1– S4E-7, Oct. 2011.

[4] Hants Kipper, Tiia Ruutmann. “Teaching for Understanding in Engineering Education”; 2012 15th International Conference on Interactive Collaborative Learning, Pages 1 – 9, Sep. 2012.

[5] Elliot P. Douglas. “Work in Progress: What is Critical Thinking?”; 2012 IEEE Frontiers in Education Conference, Pages 1– 2, Oct. 2012.

[6] Larry K. Michaelsen, Michael Sweet, Dean X. Parmelee. “Team-Based Learning: Small Group Learning’s Next Big Step: New Directions for Teaching and Learning, Number 116”. Jossey-Bass Publishers, Jan. 2009.

[7] Wen-Pin Tu, Chung-Han Chou, Shih-Hsu Huang, Shih-Chieh Chang, Yow-Tyng Nieh, Chien-Yung Chou. “Low-Power Timing Closure Methodology for Ultra-Low Voltage Designs”; 2013 IEEE/ACM International Conference on Computer-Aided Design, Pages 697 – 704, Nov. 2013.

[8] Andrew B. Kahng, Hyein Lee. “Timing Margin Recovery with Flexible Flip-Flop Timing Model”; 2014 15th International Symposium on Quality Electronic Design, Pages 496 – 503, Mar. 2014.

[9] Himanshu Bhatnagar. “Advanced ASIC Chip Synthesis: Using Synopsys Design Compiler, Physical Compiler, and Primetime”. Kluwer Academic Publishers, 2002.

[10] Chia-Ming Chang, Shih-Hsu Huang, Yuan-Kai Ho, Jia-Zong Lin, Hsin-Po, “Type-Matching Clock Tree for Zero Skew Clock Gating”; 45th ACM/IEEE Design Automation Conference, Pages 714 - 719, Jun. 2008.

Technology Enhanced Learning for Computer Science Education: A regional STEM Widening Access Case Study

Roy Sterritt, Julie Campbell, Raymond Bond, George Moore,
Giuseppe Trombino, Simon Fraser, Brian Cleland, Paul Hanna
Faculty of Computing and Engineering,
University of Ulster
Northern Ireland, UK
{r.sterritt, j.campbell2, rb.bond, g.moore, g.trombino, sg.fraser, jrp.hanna}@ulster.ac.uk

Abstract— **The paper details a regional widening access STEM initiative in utilizing computing programming for RaspberryPI, Arduino, Lego Mindstorms and Quadcopters in a Continuing Professional Development (CPD) course targeted at IT teachers from schools with a demographic that aligns with the University's widening access strategy. The paper reports on the results from a "Teach The Teachers" approach and their views of Technology Enhanced Learning for Computer Science Education before and after the CPD course.**

Keywords- STEM (Science, Technology, Engineering & Mathematics) initiatives

I. INTRODUCTION

Computers and the internet have impacted on many aspects of our daily lives, including efforts to improve teaching and learning (Hechter & Vermette, 2011; Sandholtz, Ringstaff, & Dwyer, 1997; Voogt, Tilya, & van den Akker, 2009; Williams, Linn, Ammon, & Gearheart, 2004). It has been argued that teachers have not been provided with sufficient support that extends past learning specific technology skills (Llorens, Sllanova, & Grau, 2002; Tang & Austin, 2009). The Northern Ireland Education System has often been the subject of social and political research (Goeke-Morey et al., 2012; Hewstone, Cairns, Voci, Hamberger, & Niens, 2006; Niens & Cairns, 2005). This paper reports on the beginnings of a longer term project to study the use of technology to enhance and motivate in Computer Science Education the learning experience of under-achieving classes in Northern Ireland (NI).

II. NORTHERN IRELAND EDUCATIONAL ENVIRONMENT

Northern Ireland is renowned for continuously achieving top of the educational performance league tables in terms of GCSE (16 year olds) and GCE (18 year olds) within the UK (Community Relations Council April 2014, DENI 2011/12 & National Pupil Database and Key Stage 4 attainment data 2011/12). Yet at the other end of the scale it also has amongst the worst results for under achievement of school leavers. Regionally within the UK, Northern Ireland had retained the Grammar and Secondary Schools system with academic selection at 11 years old until recently. Recently, local Government removed selection but the grammar sector are still using test operated independently by either or both of the Association for Quality Education and the Post Primary Transfer Consortium. The rest of the UK had moved to a non-selection comprehensive system in the mid 1970s. Also, for the most part, Northern Ireland has a religious segregated education system. Table 1 depicts some of the data points in a top 40 comparison between England and Northern Ireland (Community Relations Council April 2014, DENI 2011/12 & National Pupil Database and Key Stage 4 attainment data 2011/12) where achievement is measure at obtaining five good GCSE grades. Free School Meal Entitlement (FSME) vs non-entitlement (non-FSME) was used as a measure to indicate between poorer and better off backgrounds. The notable differences in the full data set is the gender gap and between those that have and have not.

A review was produced in 2001 (Northern Ireland Assembly Research and Library Service 2001), specifically discussing the gender gap and the prevention of underachievement amongst boys within the Northern Ireland education system. Since then there have been several reports and investigations on the topic of the underperformance of boys, discussing topics from Literacy and Numeracy to Community Background. More recently in 2010, (EURYDICE, 2010) revisited gender differences and concluded that boys' underachievement continue to be identified in the research. They also highlighted that solutions to this problem continually adopt a generalist approach, rather than a targeted one.

The most recent study by the Community Relations Council (2014) further identified that only Roma and Traveller children are getting poorer results than NI protestant boys from poorer backgrounds (Figure 1), in marked contrast to the fact 76% of Catholic girls from better off backgrounds are among the highest achievers. In fact in Northern Ireland the rich-poor divide in school educational achievement is stark when depicted in Figure 1. Poorer (entitle to free school meals) Protestant Boys (19.7%) & Girls (32.4%) and Catholic Boys (33.2%) & Girls (43.8%) are below the 50% mark, whereas their richer equivalents (not entitled to free school meals) are all above (richer Protestant Boys (58.6%) & Girls (71.8%) and Catholic Boys (64.5%) & Girls (76.7 %)).

Attainment gaps in England and Northern Ireland (Combined Ranked Data)		
Rank	**Category**	**Percentage achieving at least five good GCSE grades**
1st	Chinese Girls Non-FSME (Free School Meal Entitlement)	81.4
2nd	NI Catholic Girls - Non FSME	76.7
3rd	Chinese	76.4
6th	NI Protestant Girls - Non FSME	71.8
10th	Irish (in England)	66.9
11th	NI Catholic Boys - Non FSME	64.5
16th	Northern Ireland - All students	62
20th	All students - England	58.8
21st	NI Protestant Boys - Non FSME	58.6
28th	Northern Ireland Ethnic Minority	52.7
32nd	Northern Ireland Catholic Girls - FSME	43.8
35th	NI Catholic Boys - FSME	33.2
36th	NI Protestant Girls- FSME	32.4
37th	White Boys	26.9
38th	NI Protestant Boys - FSME	19.7
39th	Traveller of Irish Heritage	16.7
40th	Gypsy / Roma	9.7

Table 1 Attainment gaps in England and Northern Ireland (summary points from the top 40) [Community Relations Council 2014]

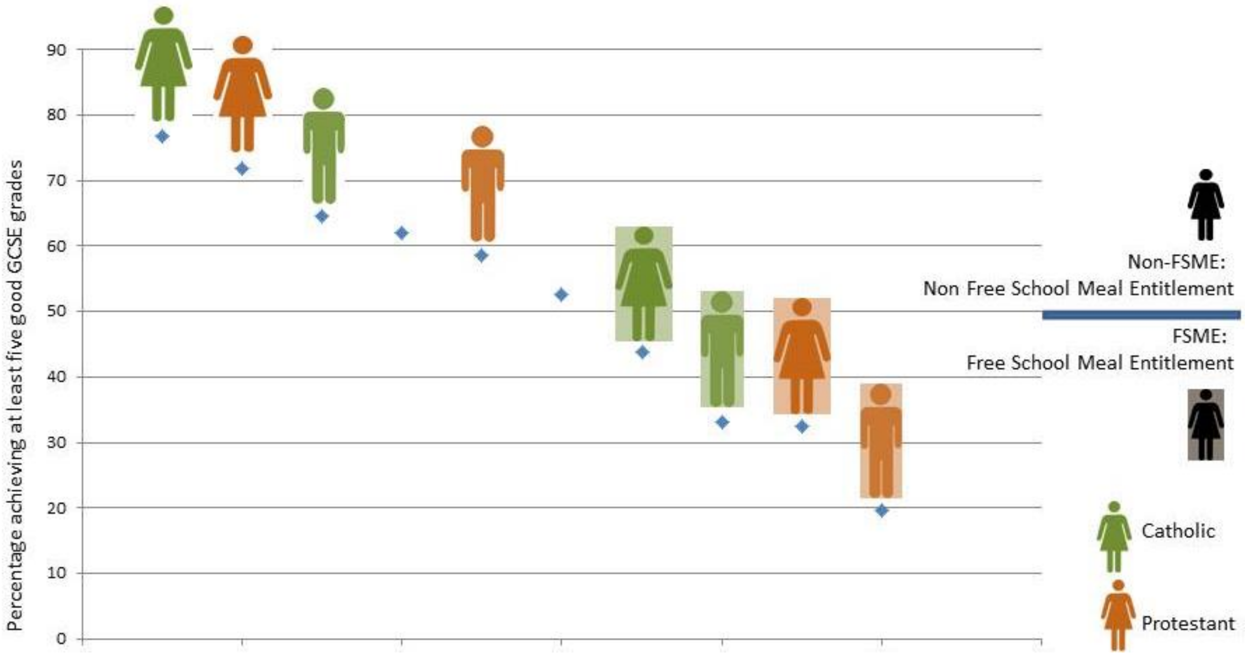


Figure 1 Attainment Gaps in Northern Ireland

III. Widening Access through Targeted Teacher Participation on specially designed CPD courses.

As a part of the University of Ulster's Widening Access initiative the University of Ulster values and corporate plan include (University of Ulster 2011):

- Raising aspirations and making our programmes accessible to those who have the capacity to benefit;
- Equality, diversity and inclusiveness;

Economically, the Northern Ireland Software sector is flourishing, yet it has a shortage of software engineers. In this widening access project, in an attempt to address the gender and religion gaps through targeting schools from the categories at the lower end of the achievement table, the team developed a CPD Teach the Teachers 'Technology Enhanced Learning for Computer Science Education' course. The course focused on introducing programming utilizing "fun" oriented hardware; namely RaspberryPi, Arduino, Lego Mindstorms and Quadcopters, as it was felt these would better engage the lower achieving male.

IV. Participating Teacher Feedback and Results

Prior to the commencement of the course participants were asked a series of questions. Some of these questions were repeated in a post course survey with an aim of establishing if their understanding and views of Technology Enhance Learning (TEL) had evolved as their knowledge and experience of TEL increased. Questions were focussed on the following areas:

- Which TEL tools were currently utilised as part of their ICT/CS curriculum for introducing programming and to what extent were they used?
- What were the factors in encouraging the development of TEL and their relative level of importance?
- What are the perceived barriers to the development and promotion of TEL tools within the next 5 years?
- What was the teachers' motivation for attending the course?

Teachers were asked about the level of use of the four main TEL tools that were to be included in the TEL4CSE course. Pre and post results varied significantly in three of the four technologies showing an increase in use in post survey results.

Findings illustrated in Figure 2 & Figure 3 show that in the use of Arduino, Lego Mind Storms and Raspberry Pi the extent of use increased in the instance 'sometimes' by a minimum of 20% form 0% in each tool.

In terms of the teachers' motivation for attending the course (Figure 4 & Figure 5), the pre to post figures for 'yes plans are already in place…' jumped from 22% to 30%; 'Yes been discussed…' jumped from 33% to 40% while the two no options dropped from 33% to 20% and 11% to 10%. From qualitative feedback (see below) it was evident in their blogs that after each of the four evenings of the course the teachers were returning to the school discussing TEL with their colleagues and planning how to integrate it into the curriculum.

In terms of the pre and post view on important factors for encouraging TEL (Figure 6 & Figure 7) only peer support dramatically changed very important dropped 13% while important increased 14% and more reliance on support from suppliers (a 7% swing from important to very important). This may indicate the teachers were becoming more confident in their own ability with the technology due to the course and would rely less on peers while seeing support of suppliers as more critical.

The teachers' perception of barriers to the development and promotion of TEL (Figure 8 & Figure 9) within their school also changed from pre to post course, for example there was a fall from 44% to 20% for lack of time (perhaps indicating again more confidence, in that it would not be as time consuming after the introduction provided by course) while lack of funds increased from 44% to 70%. This obviously became a concern after the course, and from verbal discussion during (and online blogging) it was evident the teachers wanted the equipment – so their focus may have changed to how to fund it. Lack of recognition for career development drop from 11% to 0%, while lack of opportunities to learn about the tools rose from 0% to 10%., perhaps indicating teachers wanted to learn more.

In addition to the quantitative data above some qualitative feedback that was expressed in the teachers' blogs follows:

"Just home from our first session in UUJ. Already buzzing with ideas for using simple programming apps with lower ability first years, and maybe introducing some Raspberry Pi programming with a small Yr10 group in the very near future.....Also, really looking forward to trying out the drones…… Have a feeling that our STEM team may be "persuaded" to purchase this nifty piece of kit." - Carrie

"Overall I have found this course has been quite inspiring and I cannot wait to get back into school to develop our schemes of work to try to incorporate these new technologies. If we can fire up the kids with these 'toys' I think we could get a real interest in Computer Science and that is where I want to start before I would even touch the A Level. Get them young and bring the interest through the school with them, that way when they get to A level they will already have the bug!!" - Patricia

"The work with the Lego Mindstorm kit was especially interesting and I can really see how this kind of kit would appeal to students in school, especially boys. Once again I was especially grateful for the opportunity to work with this kit in a supported environment." - Margaret

V. Conclusion

The outreach initiative has been a success to date and the data indicates that the teachers have left enthused and believe it will help with improving and motivating the pupils in the lower achievement categories. The teachers will be contacted at 6 months, 12 months, 18 months and 24 months to monitor progress. From the feedback it is also intended to offer further CPD advanced courses in each of the technologies.

Acknowledgements

Funded by the University of Ulster (2013-14) as part of its widening access and participation strategy. Thanks to the Teachers who attended the TEL CPD courses and were generous and honest with their feedback.

References

[1] DENI, Survey Of School Leavers, 2011-12

[2] EURYDICE. (2010). Gender Differences in Educational Outcomes: Northern Ireland *EURYDICE at NFER.*

[3] Goeke-Morey, M. C., Taylor, L. K., Merrilees, C. E., Cummings, E. M., Cairns, E., & Shirloe, P. (2012). Adolescents' educational outcomes in a social ecology of parenting, family, and community risks in Northern Ireland. *School Psychology International, 34*, 243-256.

[4] Hechter, R., & Vermette, L. A. (2011). *I Know I Am Supposed To, But I Just Can't: Insights into Technology Integration in Science Classrooms*. Paper presented at the Society for Information Technology & Teacher Education International Conference 2011, Nashville, Tennessee, USA. http://www.editlib.org/p/36976

[5] Hewstone, M., Cairns, E., Voci, A., Hamberger, J., & Niens, U. (2006). Intergroup Contact, Forgiveness, and Experience of "The Troubles" in Northern Ireland. *Journal of Social Issues, 62*(1), 99-120. doi: 10.1111/j.1540-4560.2006.00441.x

[6] Llorens, S., Sllanova, M., & Grau, R. (2002). Training to Technological Change. *Journal of Research on Technology in Education, 35*(2), 206-212.

[7] Mills, T., "NI education: Poorer Protestant boys underachieving", BBC NI News, 3rd Apr. 2014 http://www.bbc.co.uk/news/uk-northern-ireland-26855040

[8] NATIONAL PUPIL DATABASE AND KEY STAGE 4 ATTAINMENT DATA, 2011/12

[9] Niens, U., & Cairns, E. (2005). Conflict, Contact, and Education in Northern Ireland. *Theory Into Practice, 44*(4), 337-344. doi: 10.1207/s15430421tip4404_7

[10] Northern Ireland Assembly Research and Library Service (2001). Selected Current Research in Gender and Educational Attainment *Research Paper 06/01*.

[11] Sandholtz, J. H., Ringstaff, C., & Dwyer, D. C. (1997). *Teaching with technology: Creating student centered claddrooms.* New York: Teacher College Press.

[12] Tang, T., & Austin, J. (2009). Students, perceptions of teaching technologies, application of technologies, and academic performance. *Computers & Education, 3*(54), pp. 1241 - 1255.

[13] University of Ulster, Corporate Plan Professional Education for Professional Life 2011/12 to 2015/16, http://www.ulster.ac.uk/corporateplan/documents/CP.pdf

[14] Voogt, J., Tilya, F., & van den Akker, J. (2009). Science teacher learning for MBL-supported student-centered science education inthe context of secondary student educationin Tanzania. *Journal of Science Education and Technology, 18*, 428-429.

[15] Williams, M., Linn, M. C., Ammon, P., & Gearheart, M. (2004). Learning to teach inquiry science ina technology-based environment: a case study. *Journal of Science Education and Technology, 13*(2), 189-206.

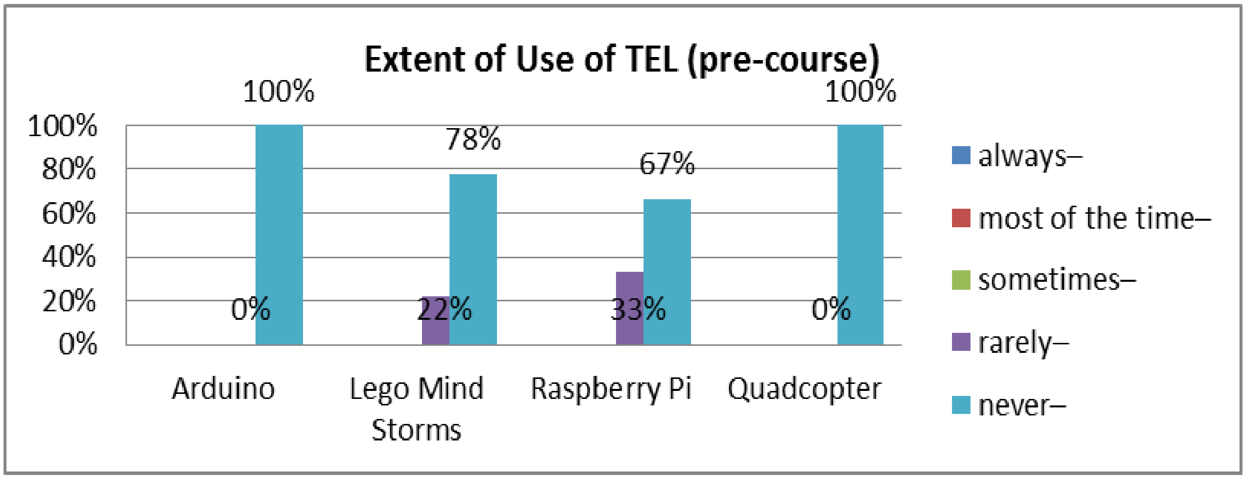


Figure 2 Pre-course – extent of use of TEL in School

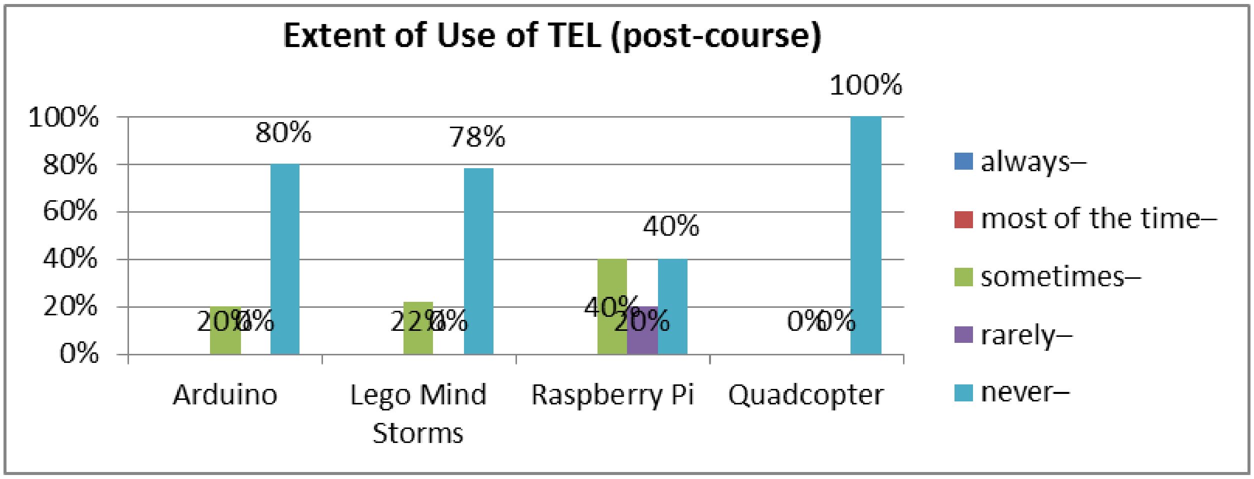


Figure 3 Post-course extent of use of TEL

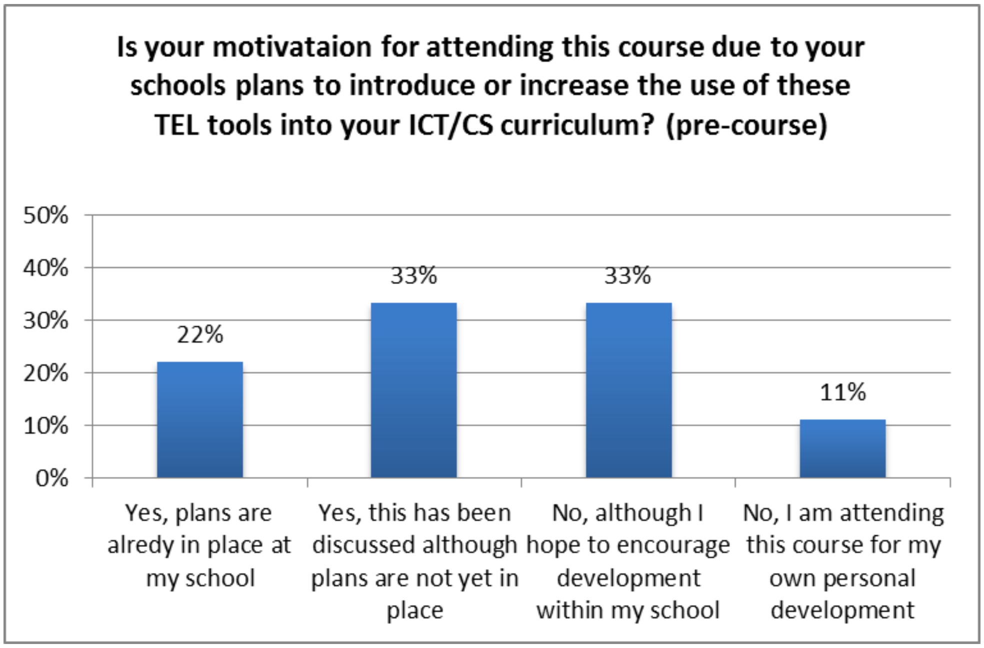


Figure 4 Pre-course motivation to use TEL

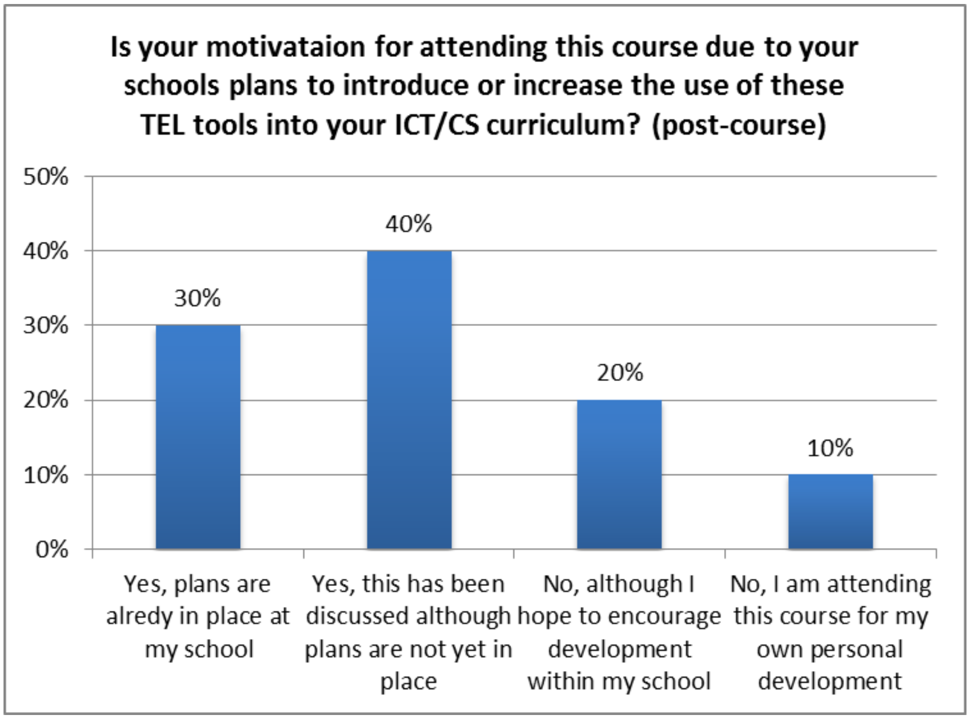


Figure 5 Post-course motivation to use TEL

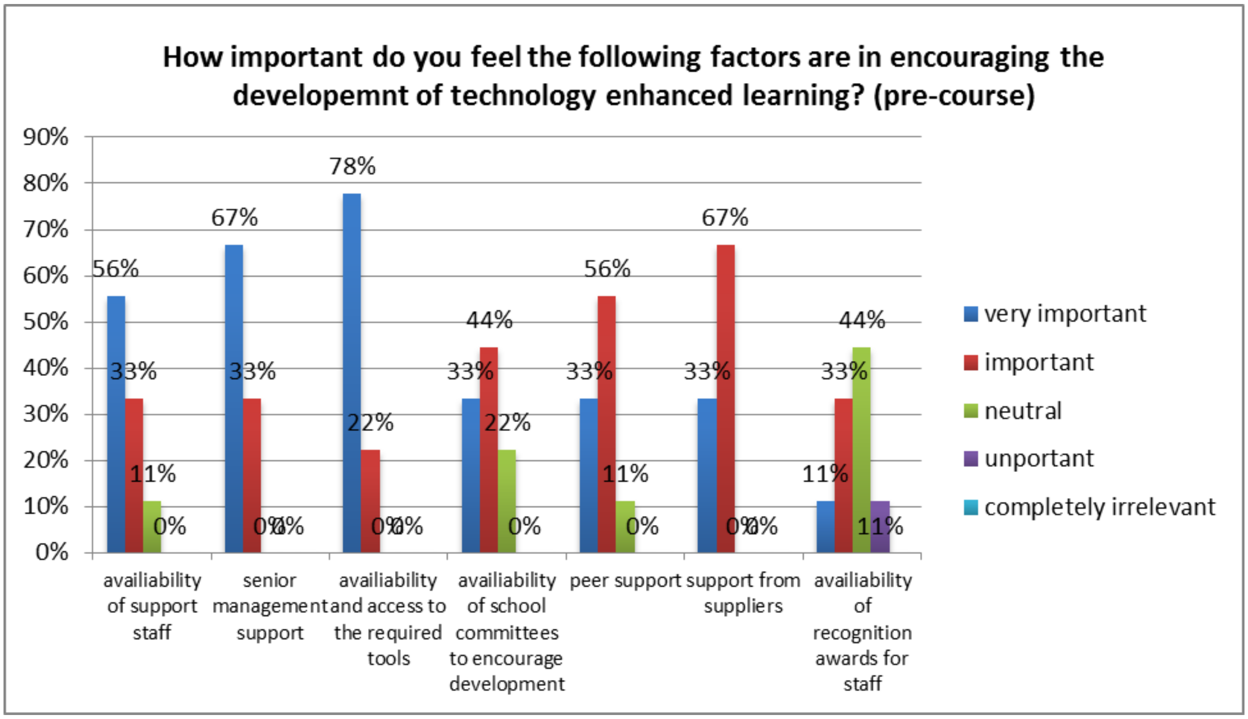


Figure 6 Pre-course Factors influencing developing TEL

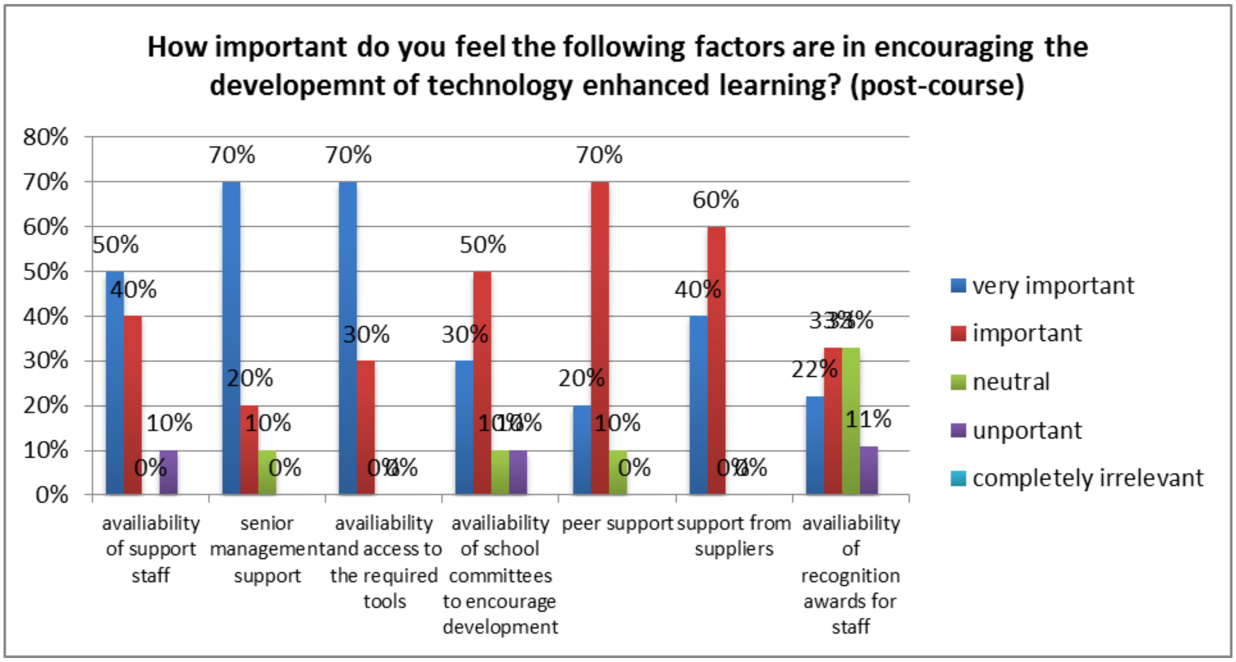


Figure 7 Post-course Factors influencing developing TEL

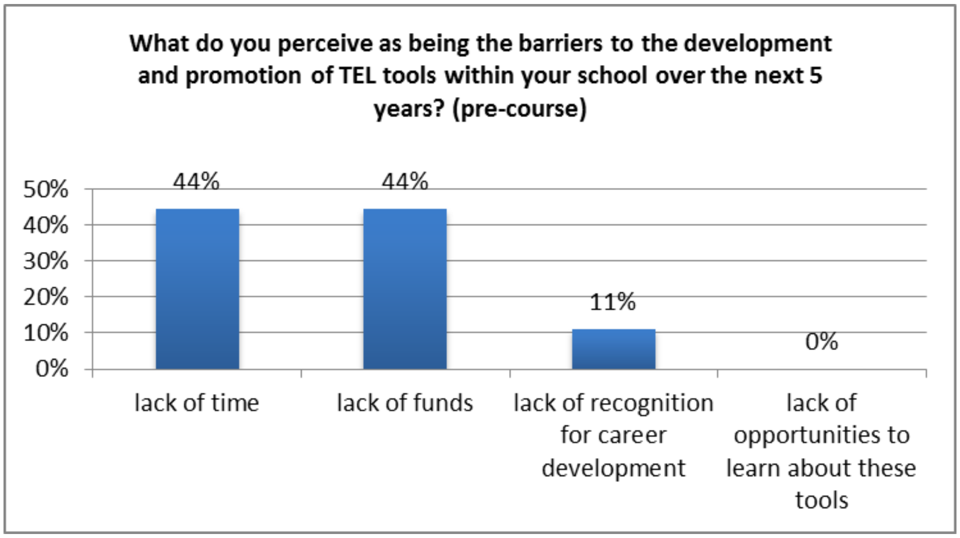


Figure 8 Pre-course Barriers to developing TEL

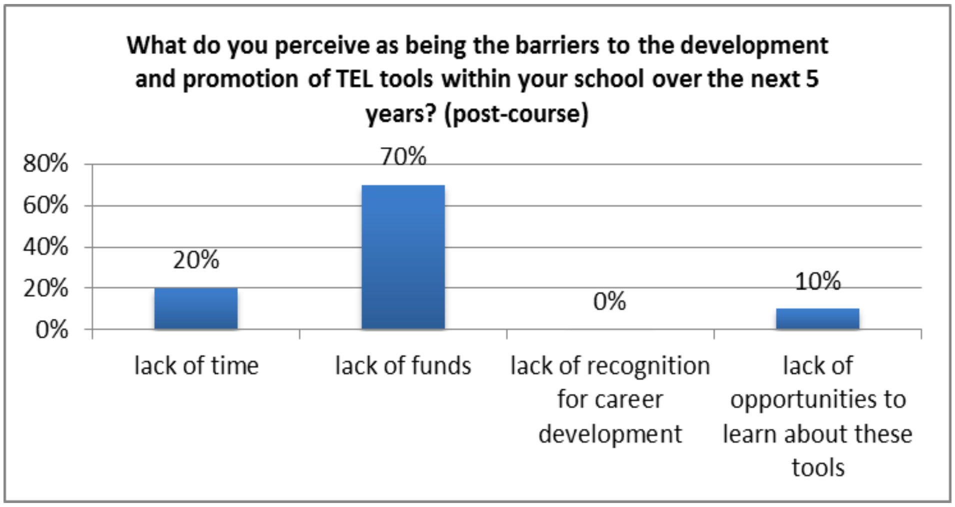


Figure 9 Post-course Barriers to developing TEL

The Development of Interactive Distance-Learning Laboratory for Teaching Digital Design in Electrical and Computer Engineering

Bassam Shaer and Mohamed Khabou
University of West Florida, bshaer@uwf.edu, mkhabou@uwf.edu

***Abstract* – Distance education provides an avenue for students far from a university center to complete their degrees in a timely fashion. This paper describes a distance learning (DL) laboratory setting in which students at remote locations synchronously engage with an instructor and students at a university center and complete laboratory experiments in a digital design laboratory course offered by an ABET-accredited electrical and computer engineering program. The Polycom-based DL system supports two-way, real-time communication, which includes audio, video and lecture material, between multiple locations. The required DL equipment is described and the unique challenges faced by both students and faculty in this environment are discussed.**

Index Terms – Engineering distance education, Interactive distance learning studio

Introduction

This paper is a reflection of our experience in developing, piloting, and delivering two interactive synchronous distance-learning (DL) labs at the University of West Florida (UWF). UWF offers ABET accredited programs in electrical engineering and computer engineering [1]. Numerous authors have investigated different distance learning environments [2-10]. UWF, with a student body of approximately 13,000, has its main campus location in Pensacola (PNS) but has a strong presence in areas to the east through its Emerald Coast branch campus locations. The largest of these branch campus areas is located in Fort Walton Beach (FWB) due in large part to the proximity of Eglin Air Force Base and associated contractor companies that require a highly skilled workforce. Many degree programs offer courses at UWF Emerald Coast locations (e.g., criminal justice, teacher education, hospitality management, computer science, etc.), but, due to a unique delivery method described in what follows, the engineering department offers a complete degree program, which includes all required laboratory courses.

Approximately eleven years ago and primarily due to the needs of the Air Force as well as supporting contractor companies, UWF began offering its electrical engineering and computer engineering program in Emerald Coast using a synchronous distance education model. At the outset, no faculty were located in FWB while at the present time, three full time engineering faculty out of the faculty body of eight are resident in Emerald Coast.

All lecture courses in the UWF engineering program are offered in an Interactive Distance Learning Studio (IDLS) and not in an increasingly common asynchronous, self-paced learning environment [11, 12, and 13]. The UWF [14] IDLS, which employs commercially available equipment from AMX [15], and Polycom [16], uses a dedicated Internet connection which reserves 3Mb/s for real-time audio and video and an additional 3Mb/s for data. In this setting, an instructor from either campus, with the help of facilitators at both locations, simultaneously delivers a lecture course to students in Pensacola and Emerald Coast. The real-time audio and video connection allows the faculty member to see, hear and interact with students at both ends with the same capability afforded to the students. Using a Tablet PC or a Wacom Cintiq 22HD, the data connection allows the instructor to present lecture notes via an electronic whiteboard or PowerPoint as well as to utilize engineering software through use of a commercially available projection system in both classrooms. The system also allows lecture audio and data to be recorded so that students can review lecture material at a later date.

While faculty lectures are presented to students in both locations, faculty at either end were responsible for teaching laboratory sections in their corresponding location in a traditional face-to-face setting. This required the acquisition of identical --and often costly-- laboratory resources at both locations and the necessity of offering two lab sections even if enrollment is low at either location. This exact situation presented itself many times especially in our upper level digital design lab which is required only for our computer engineering students. In addition, students taking the co-requisite labs of many classes don't have direct access in the lab to the professor on the "far" side who teaches the lecture portion of the class. This leaves the students talking the lab at the "far" side at a disadvantage. This situation sparked the idea of developing a distance-learning laboratory to alleviate these problems.

After securing funding though UWF's Instructional Technology Enhancement Projects (ITEP), the authors researched, acquired, and installed the necessary hardware and software to create an interactive distance-learning lab and piloted the offering of the digital design lab in such a setting. In this paper, we discuss the details of the lab setting and the challenges that students and faculty faced the first time this setting was utilized.

DISTANCE LABORATORY SETTING

The authors established two interactive distance learning laboratories (one at Pensacola campus and the second at the Emerald Coast campus) that incorporate the latest in computer-aided design (CAD) software and allow students at one campus to interact with their professor and classmates at the "far" campus. The two interactive distance learning laboratories are hosted in the regular digital design labs at both ends and did not require extra lab space.

Like the distance learning classrooms we use to deliver lectures, the interactive distance learning laboratories are setup with video conferencing equipment that allows the students and professors on the "far" and "near" sides to interact in real time. In addition, the setup includes computers with the latest CAD software for designing and testing digital and electronic circuits. The computers are setup to allow remote access to the CAD software from any computer with an Internet connection. This, for example, will allow a student to connect to the host computer, design his/her circuit, simulate it, and discuss it with the professor without having to be physically in the lab.

The CAD/simulation software we decided to use is Multisim made by National Instruments, a pioneer company in circuit simulation and testing. In recent years there has been a shift towards a "bench-like" software that simulates in real-time the workplace of an electronic designer. Such software produces a computer-based model of circuits and uses virtual electronic instruments (function generators, oscilloscope, power supplies, voltmeters, etc.) to test these circuits. Multisim is considered by many to be the industry standard (Figure 1).

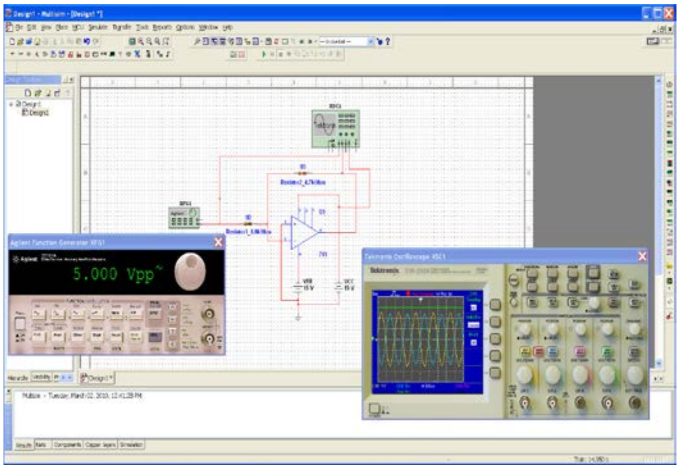

Design1 - Multisim - [Design1 *]
Agilent Function Generator XFG1
5.000 Vpp
Tektronix Oscilloscope XSC1

Figure 1. Screen capture of the Multisim software.

A detailed list of equipment for both labs is detailed in Table 1. As can be seen in Table 1, the total equipment coast including hardware and software was less than $58K (which includes a backup unit). Installation costs are not included since the authors installed the equipment themselves with the help of engineering students for whom this was a learning experience in hardware and software installation.

Table 1. Price list of acquired project items

Item	Cost
HDX 9004 XLP Polycom with EagleEye Camera	$10,798
HDX 9004 XLP Polycom	$3,500
EagleEye Camera	$2,450
Camera Mounting	$516
HDX Ceiling Microphone Array	$1,718
LG 55" HD TV with Wall Mount and Warranty	$11,574
Bose Sound System/Amplifier	$1,594
Cables, Wires, and Adapters	$2,429
Wacom CINTIQ 21UX Interactive Pen Display	$9,996
Ceiling Mount	$386
Ploycom Touchscreen Control	$1,151
Mobile TV Cart with Shelf	$788
Dell Computer T7500 & OptiPlex 9010	$4,617
Misc Polycom Parts	$181
Microsoft Surface RT Tablet	$988
Microsoft Surface Pro Tablet	$1,422
HD Projector	$3,558
Total	**$57,666**

Figure 2 shows the wiring diagram of all needed equipment in both labs. Each IDLS uses a Polycom HDX 9004 XLP system configured as shown in Figure 2 with two EagleEye HD cameras and two 55 inch HDTVs for students and instructor views. It uses a high fidelity Bose Sound System, one HDX ceiling microphone array to capture students' questions and one podium microphone for the instructor. The instructor's podium has a high-end computer connected to a Wacom CINTIQ 22HD Interactive Pen Display that can be used as a white board and/or to annotate lecture notes. The screen content is displayed on projectors at the two lab locations.

The equipment in the new labs is compatible with the different DLL settings at other UWF classrooms and has the capability to connect to multiple locations at once (the new system is capable of connecting up to 8 sites simultaneously). Figure 3, shows a screen capture of four simultaneous connections to four different locations. The left top corner shows an older version of Polycom 9001 system, the right top corner show the new Tandberg system, the bottom left corner shows a connection using a PC computer system using Telepresence m100 PC software from Polycom while the right bottom view is a Polycom 9004 system.

The new equipment also allows an instructor to teach his/her class/lab from his/her office or from anywhere in the world to multiple sites at the same time. Figure 4 and Figure 5, show one of the authors presenting lecture material to two different locations from his office's computer. The system can also be used as a multipoint videoconferencing system for meetings/presentations and interaction between students and instructors.

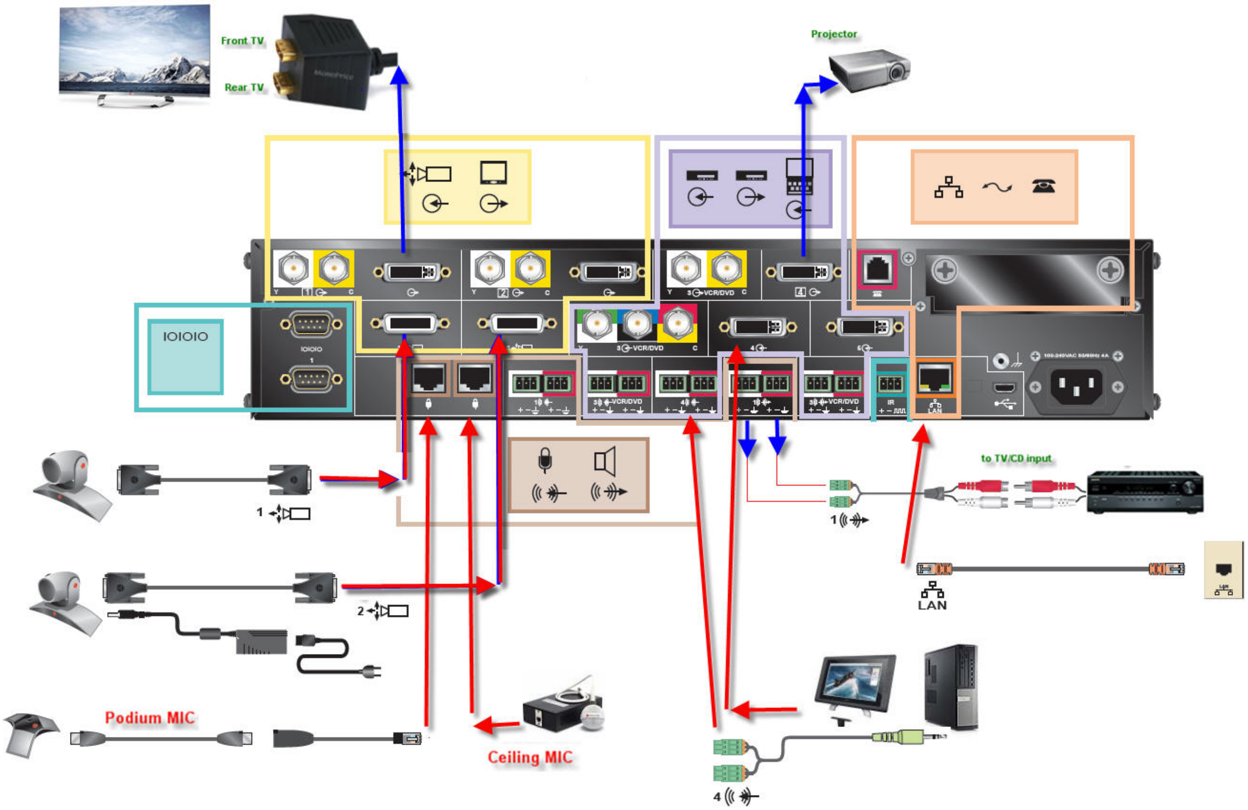
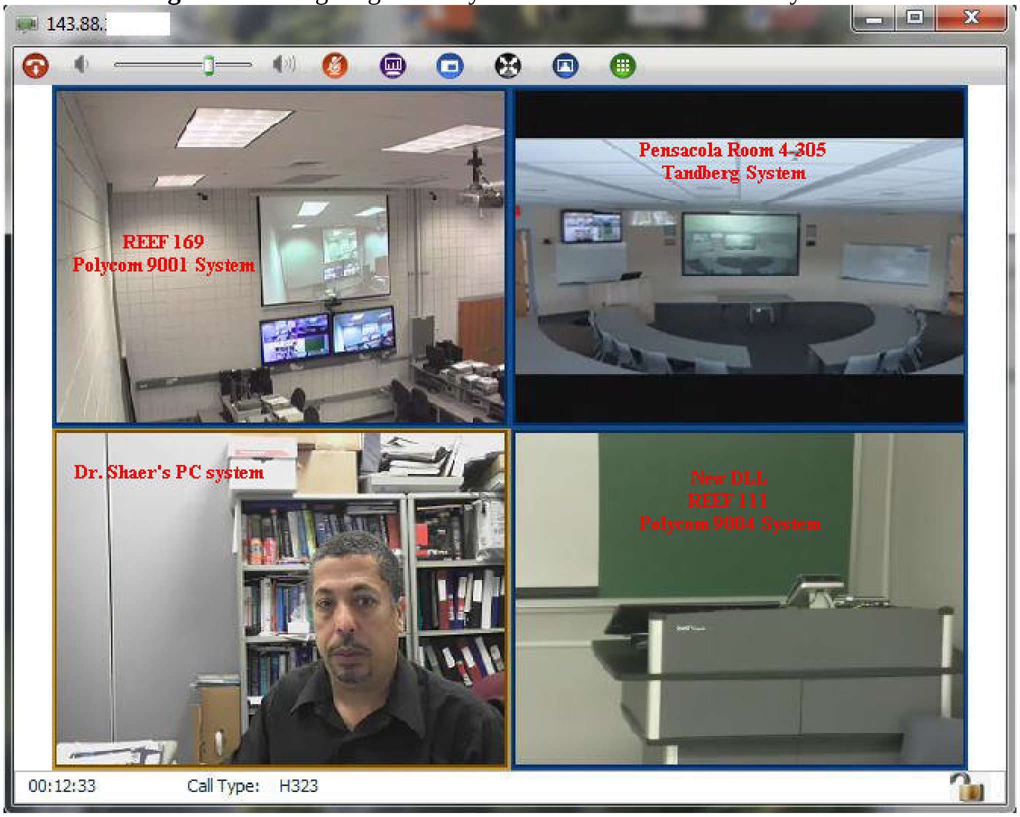


Figure 2. Wiring diagram of synchronous distance laboratory model



00:12:33 Call Type: H323

Figure 3. Four Simultaneous Multipoint Connections

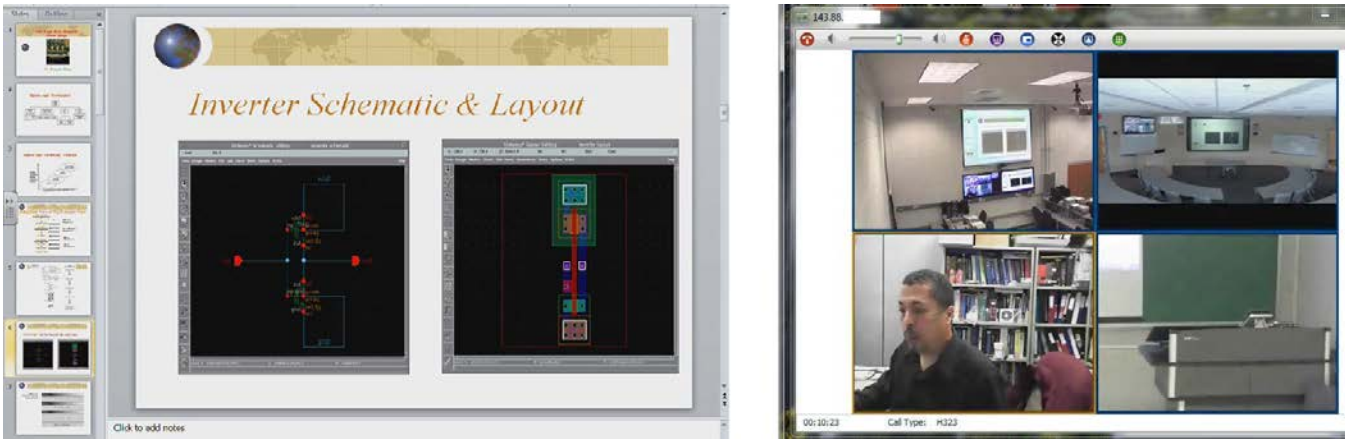


Figure 4. Office Presentation to Multiple Rooms

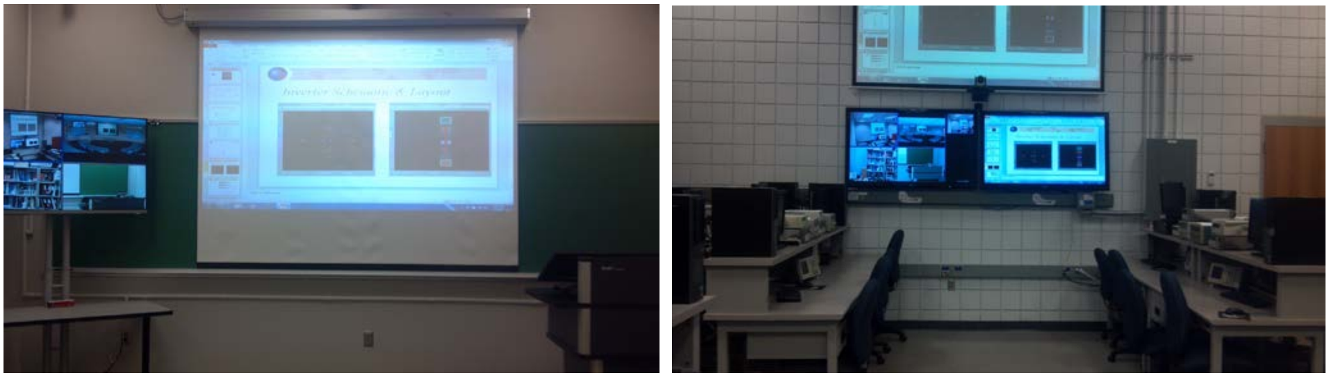


Figure 5. Students' View

STUDENT LEARNING IMPACT

The digital design course (EEL4712) is required for all computer engineering students and is commonly taken as an elective by many electrical engineering students. Currently, the class is organized as 3 hours of lecture and 3 hours of lab every week. The advances in computer technology in the areas of computer software and hardware have paved the road for instant results for circuit simulations. The new CAD tools stimulate the senses with colorful, accurate circuit waveforms in graphical and tabular formats. No longer must the user enter hand-coded data for circuits or systems; a click of the computer mouse is all it takes to create a professional-looking schematic of a circuit or a block diagram of a system. The simulation is fast and the results are many times better than hand calculations could provide. Circuit simulation complements the laboratory environment, but cannot replace it entirely. Skills acquired in the laboratory allow the ECE students to examine circuits from a different viewpoint.

The digital course is reorganized into blocks of time when students can interact with the instructor in real-time as new material and concepts are presented and then work in small teams (typically two-student teams) to design, simulate, synthesize, and test digital circuits. Students are initially trained on the use of the equipment the first week of class. Hands-on learning, with adequate documentation, is often the best way to proceed. After the students can operate the equipment and the CAD tools, they start the design component. Working with the instructor and with their classmates, they learn how to do efficient circuit/system design on different levels of abstraction. The most effective way to complete this stage is by instructor-guided learning. When an instructor presents new material, students are able to download circuits and simulate them in real-time to have a better intuitive feeling of what is happening with the circuit than in cases where simulation was not performed. By the end of the course, students are able to design their own circuits, share their results with fellow students and give oral presentations.

The simulation, verification and testing of the designs will be done using the Electronics WorkBench/Multisim software and the Quarus II as CAD tools. Circuit simulation is a very powerful method for solving many types of complicated digital circuits and systems. The Electronics WorkBench software package offers easy-to-use graphical user interfaces and provides fantastic graphical output that rivals the best electronic bench equipment. Virtual instruments are cheap to own and operate and they never need service and calibration. Students enjoy simulating circuits without having to put up with the hassles of the physical networks and the vast number of interconnections that link everything together. This encourages students to experiment with circuit components, identify the component's function and link each function to the characteristics of the circuit's response. The more time spent studying a topic from different viewpoints with various techniques, the better the topic will be understood.

DISTANCE EDUCATION FACULTY-STUDENT CHALLENGES

While the above described synchronous distance education model is preferable to an asynchronous one, there are many challenges that are not present in a typical laboratory setting. The major challenges include the following:

- Students in the distant laboratory are only visible on large screen monitors located at the front (primarily for use by students) and rear (for use by faculty) of the DLL. As such, establishing a connection with students at the distant end and using their facial expressions for feedback during lecture to gauge understanding is extremely difficult. The lack of direct interaction between faculty and students at the distant end manifests itself in end of semester evaluations of the laboratory and instructor where the distant end typically has a less positive indication of the course and instructor when compared to the near end.
- Distant student lab reports are submitted electronically for grading using eLearning drop boxes by scanning a paper copy or by creating their work directly on a Tablet PC by utilizing Windows Journal, Microsoft Word or Microsoft OneNote. However, these electronic documents are much more time consuming to grade than paper copies.
- Interacting with distant students during office hours to clarify problems is challenging. While a telephone conversation or an email especially with a handwritten attachment created on a Tablet PC can be utilized, there is again a lack of personal interaction.
- It is almost impossible to debug a circuit wiring diagram or equipment malfunction over the distant site.

As a final comment, in general, the time spent on a one credit synchronous distance education lab for both the student and the faculty is more than a typical three credit lecture course. Faculty are cognizant of the additional effort and take this into account in terms of being more flexible than would typically be the case. However, the same recognition is not afforded to faculty in any distance education program at UWF.

STUDENT LEARNING OUTCOME ASSESSMENT

The evaluation of the new DLL and its impact on student learning is a major element of this work. Assessment was conducted using an assortment of methods to provide immediate and long-term feedback. The following methods were used:

- *Student performance on course examinations*: This most fundamental measure of student performance provides immediate feedback to instructor to allow adjustment of instructional techniques using the interactive lab environment.
- *Student responses on course evaluations*: Standardized course evaluations were collected from students at the end of each semester for each course using the interactive lab. We focused on the responses to questions most clearly related to the new teaching environment. In addition to the standard questions on the form, we added additional questions designed specifically to assess the effectiveness of teaching in this interactive environment.
- *Student preparation for subsequent courses*: This assessment technique measures perhaps the most important aspect of how well the new instructional format educates students. By comparing the preparation level of students who have learned digital design in the new environment with that of students learning under traditional structure, the value of the interactive laboratory environment can be assessed.

In the first year of offering digital design over the new DLL laboratory, students' evaluations were very negative for several reasons. "Instructor does not have long hand to reach the distant site to move a wire or to adjust a function generator or a power supply or to debug a bad wire connection." This negative response was removed by hiring an undergraduate teaching assistant who is familiar with the equipment and has taken the course to be located at the distant site to be the hand of the instructor.

SUMMARY

This paper has described the development, setting of distance-learning laboratory, and the challenges faced by faculty and students. The facilities currently utilized have been described, and the challenges faced by faculty and students have been noted.

It is expected that the previously noted challenges will be prominently noted by students as responses to the questionnaire. The student learning outcomes assessment, taking into account student location must still be completed. If student learning outcomes are relatively similar between both locations, then the UWF distance education delivery model will be validated. If the outcomes vary significantly among the two student populations, then serious discussion with all faculty will be required in order to fix deficiencies.

REFERENCES

[1] University of West Florida, Department of Electrical and Computer Engineering. Available: http://ece.uwf.edu/

[2] Carnevale, D. 2000 "Turning Traditional Courses into Distance Education". Chron. High Educ., vol. 46: pp. A37-38.

[3] Mirakian, E. and L. Hale. 2007. "A Comparison of Online Instruction vs. Traditional Classroom Instruction in an Undergraduate Pharmacology Course", Proceeding of the 3rd Annual GRASP Symposium, Wichita State University, pp. 95-96.

[4] Blackwood H. and C. Trent. 1968. “A comparison of the effectiveness of face-to-face and remote teaching in communicating education information to adults.” Manhattan: Kansas State University, Cooperative Extension Service. (ERIC Document Reproduction Service No. ED 028-324).

[5] Jaeger, M. 1995. “Science teacher education at a distance.” The American Journal of Distance Education, vol. 9 no. 2, pp. 61-75.

[6] McCleary, I. D. and M. W. Egan. 1989. “Program design and evaluation: Two-way interactive television”. The American Journal of Distance Education, vol. 3 no. 1, pp. 50-60.

[7] Molnar, A. R. 1997 “Computers in education: A brief history.” Technology Horizons in Education Journal, vol. 24 no. 11, pp. 63-68.

[8] Souder, W. E. 1993. “The effectiveness of traditional vs. satellite delivery in three management of technology master's degree programs”. The American Journal of Distance Education, vol. 7 no. 1, pp. 37-53.

[9] Thomerson, J. D. and C. L. Smith. 1996. “Student perceptions of the affective experiences encountered in distance learning courses.” The American Journal of Distance Education, vol. 10 no. 3, pp. 37-48.

[10] Spooner, F. et al. 1999. “Student Rating of Instruction in distance learning and on-campus classes,” J Educ Res. Vol 92, pp. 132-140.

[11] Blackboard Inc. Available: http://www.blackboard.com

[12] ANGEL's web-based teaching and learning tools. Available: http://www.angellearning.com

[13] Desire2Learn Incorporated, 1999-2009. Available: http://www.desire2learn.com

[14] Shaer, B. and A. Fuchs. 2008. “Work in Progress: Student Learning Outcomes in a Distance Education Setting” The 38th ASEE/IEEE Frontiers in Education Conference, pp. 12-13.

[15] AMX hardware and software solutions. Available: http://www.amx.com/

[16] Polycom, Available: http://www.polycom.com/

[17] Center for University Teaching, Learning, and Assessment (CUTLA). Available: http://uwf.edu/cutla/index.cfm

An environment to promote a visual learning of Calculus

P. Salinas[1], **E. Quintero**[2], **and E. González-Mendívil**[2]
[1]Mathematics, ITESM, Monterrey, Nuevo Leon, Mexico
[2]Mathematics, ITESM, Monterrey, Nuevo Leon, Mexico

Abstract - *This paper presents a didactical experience taking place at 2012 in a Calculus I course with 24 college students. The scenario, including the simulation of uniformly accelerated motion over a straight line, is analyzed through a dynamical software. Additional to the simulation, SimCalc educational software allows the graphical interplay between velocity and position graphs, and through this visual perception, the emergence of prediction questions about the motion are promoted. Students should deal with them making use of algebraic and numeric procedures related to linear and quadratic equations. Results make us believe in the potential for the learning of calculus of a new way to deal with the visual perception of graphs, as tools for the mathematical reasoning involving numerical, algebraic and graphical representations of functions involved with the visualization of the motion scenario.*

Keywords: Calculus, learning, technology, innovation.

1 Background

As part of a research group working at an educational institution in northern Mexico, our claim since 2000 has been to offer students an innovative approach to the teaching and learning of calculus. We focus on the learning of mathematical knowledge favoring the practical application students will find in later engineering courses.

Based on research we decided to declare the 'practice of predicting the value of a magnitude that is changing' as a 'conducting thread' to reorganize the calculus curriculum. Our approach integrates the ideas of rate of change and accumulation in order to predict this value.

We propose studying motion on a straight line as the first scenario where the magnitudes position and time appear related. We decided this because of Newton's origin of calculus and the familiarity a student could have when thinking about this phenomenon. Through this scenario, students acquire a first impression of calculus concepts and procedures.

Traditional curricula separate differential and integral calculus courses because of the logical structure that traditional textbooks handle. But the reorganization we have been addressing through the prediction practice unifies both courses. This approach to calculus has been discussed and shared in several forums, as well as in educational journals, particularly in [1] and [2].

Through the publication of textbooks and their use in our institution, reflection on the way it impacts the classroom dynamic is our daily source for research in order to strengthen our work. Engineering programs offer the first calculus course with "Applied Calculus: Mathematical competencies through contexts" [3].

Within this innovative approach, we have been working with different software and analyzing its contributions to visual learning. Particular features in technology products do matter for the learning proposal to be strengthened. We find in technology the opportunity to create a way to interact with different representations of mathematical knowledge and carry it into the classroom. We act in accordance with the educational trend [4] named about the role and use of technology: to support an experiential approach to learning.

SimCalc MathWorlds (http://www.kaputcenter.umassd.edu/products/software/), hereon called 'SimCalc', offers dynamically manipulable Cartesian graphs that are linked to motion simulations. Since its conception, James Kaput (1942-2005) was addressing the semiotic perspective of mathematical notations for education; he provided us with a framework for understanding issues that emerged when we attend to the way mathematics comes to be known through the act of teaching and learning within classrooms. SimCalc projects share the common goal to giving students the opportunity to access the important mathematical ideas of change and variation.

SimCalc allows designing of activities in which the goal can vary according to the purpose of the researcher. Particularly, we are concerned on a didactical sequence of activities to create a learning environment in classroom to develop the graphic visualization of the behavior of a function through its derivative. In this paper we want to share the results of the study of a pilot activity regarding the presence of SimCalc in a Calculus I course during spring 2012.

2 Rationale

You can easily recognize in calculus' students the cognitive difficulty [5] identifies dealing with the flexibility of representations. Going back and forth between visual and analytic representations is a matter widely recognized at the core of understanding mathematics. [6] clarifies the notion of representation with underlying cognitive aspects that may be sources of difficulty in learning mathematics. Representations are also signs associated in complex ways, which are produced in compliance with rules to form a system. Like language, they are tools to produce new knowledge as the result of operations and the organization of cognitive structures in our mind.

[7] provides a key insight to analyze the cognitive processes involved in mathematical thinking: you have several systems of representation (algebraic, numeric and graphical) that must be coordinated during mathematical activity. Duval identifies two types of transformations that can be done on those semiotic representations, named as treatment and conversion. The first transformation (treatment) refers to the changes made within the same representation, while the second (conversion) refers to the ability to change representation, including transformation of linguistic statements into numeric, algebraic or graphical.

It is important for calculus students to have a fluent, coordinated and simultaneous use of algebraic symbols, graphics, and graphical and numerical patterns. Currently software is a powerful tool to 'instantly' show different representations requested. But even when "the most obvious action is to show several possible representations at the same time" [8] (p. 11), the goal of an appropriate manage of mathematical representations is not immediately attainable.

[8] has recognized different levels for cognitive processes ranging from a recognition process by association that works at a superficial level of the semiotic representation, to a cognitive process that demands some kind of double discrimination acting on different semiotic representations. Activities relying in a mere juxtaposition of multiple simultaneous representations of the same mathematical object promote cognitive processes of the first level, bringing limitations for the learning process.

It is natural to describe a symbol as something that takes the place of another thing. [9] analyzes evolution of symbolic thinking over time. They state that a system of external symbols could become a meta-cognitive mirror in the sense that one's ideas about some field of knowledge can be socially shared; "then we can see our own thought reflected in that "system" and discover something new about our own thinking" [9] (p. 101).

Using computers could transform mathematical knowledge. [10] has been working on this matter looking for new theoretical and methodological tools to enlighten the learning process related to technology integration. By means of a situated abstraction, [10] refers to a way in which a community of students can develop a common discourse and coincide with their teacher about how they are talking of the same mathematical abstractions. It could happen that different meanings are associated with the abstraction, but this benefits the learning situation because differences are explicit to reflect and discuss, and to promote them becoming associated with the teacher's ones. This way of thinking makes the student's expressions gain some mathematical legitimacy "even if they differ substantially from traditional mathematical discourse" [10] (p. 2).

[11] addresses the issue of conceiving new ways to think on students' meanings. Students are so familiar with computational tools, that eventually, the exploration allows them to reorganize strategies for problem solving. Through an exploring experience they can make some situated remarks within the computer environment. Those observations may relate to a certain property, theorem or formula, where the environment facilitated the identification. This comprises a situated proof, the result of a systematic exploration purposely exploited inside a computational environment in order to "prove" mathematical relationships.

The continuous dynamic media offers a special feature that allows the promotion of co-action between the student and the tool. [12] refers to this concept describing, "how the user of a dynamic environment guides the action upon environment and is guided by the environment as a fluid activity" (p. 505). This dual process of guiding and being guided, points out the intentional attitude when carrying out a human activity.

Duval's framework gives a convenient way to analyze the introduction of technology in classroom. According to our calculus course, we ask technology to foster the possibility to relate graphical information and language used by teacher and students when dealing with a motion situation. We opt to focus on make students familiar with the interpretation of graphs. This goal certainly needs time to be under cognitive control, but we found it relevant to accomplish, since the other semiotic representations (numeric and algebraic) could be considered in order to articulate thoughts and enrich the cognitive process.

3 The activity designed in SimCalc

Favoring graphics in calculus should promote a visual learning. This intention is fulfilled using SimCalc software that brings mathematical expressivity in the classroom through a simulation where a character performs a motion animation according to certain conditions. Through a SimCalc document we can design a visual scenario on the computer screen to support the cognitive process that takes place when a didactical activity is set. There we can handle the position-time graph and the velocity-time graph in different coordinate systems. At the same time, we can visualize the motion animation that is being performed by the character, and bring to the classroom the prediction of character's position as the magnitude to analyze.

The experience we share here took place in 2012 spring semester during Calculus I course in charge of one of the authors. The 24 students participated bringing to class their own laptop where SimCalc software was installed. The teacher made the introduction of SimCalc before, when studying uniform motion. There, teacher interacts with the classroom computer, and the group visualizes the display in front to discuss what the character (named Ryan) should do. In a one-hour session the scenario designed with motion at a constant velocity was analyzed; character moves over the horizontal straight line according to the information that velocity and position graphs describe visually. Figure 1 shows the scenario where co-action, situated abstraction and proof, treatment and conversion take place, bringing the opportunity to analyze the learning process emerging. The interaction with SimCalc is dynamical; it is made through the hotspots, black dots over the graphs. Dragging the hotspots allow changes in velocity graph having in response an action over position graph according to the Calculus Fundamental Theorem, part of the infrastructure of the software.

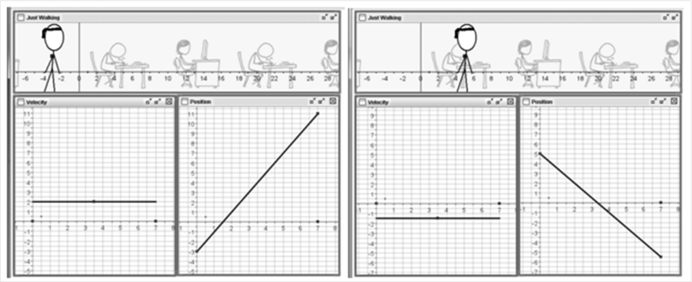


Fig. 1. Two SimCalc documents showing a uniform motion with constant velocity.

The algebraic representation of uniform motion is generalized in class through the identification of several examples with particular numbers involved. The teacher proposes the examples through SimCalc, the specific numeric values are visualized by the graphs (initial values and slopes). The scenario invites students to relate the constant function

$$v(t) = v_0 \qquad (1)$$

with the linear function

$$x(t) = x_0 + v_0 t, \qquad (2)$$

algebraically and visually. There is identified also the algebraic process of calculating the derivative (from x(t) to v(t)) or the antiderivative (from v(t) to x(t)) between the two functions.

After having studied uniform motion, the textbook provides a numeric strategy and motion arguments in order to relate algebraically the linear velocity function

$$v(t) = v_0 + a_0 t \qquad (3)$$

with the quadratic position function

$$x(t) = x_0 + v_0 t + (1/2)a_0 t^2 \qquad (4).$$

Velocity function is identified as the derivative of position, and position function is identified as the antiderivative of velocity. After this, SimCalc brings the opportunity for the graphical interplay with uniformly accelerated motion, the activity we like to share here.

The activity Ryan MUA (*Movimiento Uniformemente Acelerado*) takes place in one-hour class having 2 parts. First, students access the Ryan MUA SimCalc document through the technological platform where the course is distributed. The document shows the same design as in uniform motion, where we privilege showing velocity and position graphs, and the animation of motion performed in a horizontal line. Figure 2 shows two images for linear velocity and the corresponding position graphs.

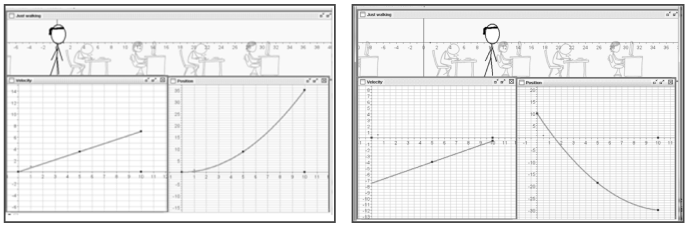
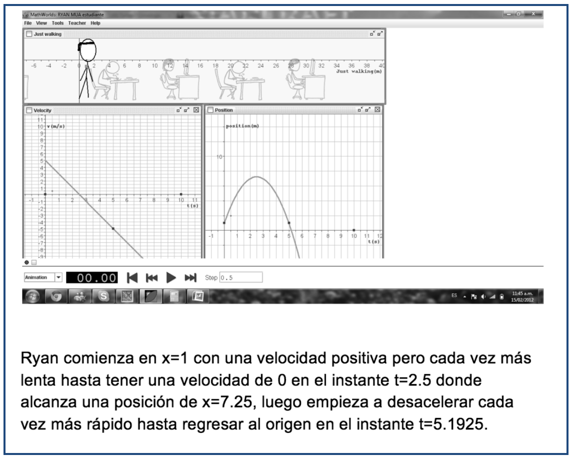


Fig. 2. SimCalc document performing a motion with linear velocity.

The hot spot on the middle of the velocity segment in Figure 2 allows the dragging of velocity segment up and down, and the hotspot at the end of velocity segment allows the dragging for the change of slope. Through the exploration, students get to know about the behavior of position graphs in terms of the behavior of linear velocity. They are encouraged to interact with SimCalc and between each other, and simulate four kinds of motions: to the right increasingly faster (or slower), and to the left increasingly faster (or slower). After half an hour, we propose an image at the screen in front; it is a velocity graph, a linear segment with initial value 5 meters per second on the vertical axis, and intersecting horizontal axis in time 5/2 seconds. We encourage students to interact with Ryan MUA SimCalc document and construct a motion simulation that suits the velocity given. They have to consider that Ryan´s initial position is 1 meter. Activity finishes when they deliver a word document with their picture (screenshot) of SimCalc, the algebraic representation of the motion, and with the interpretation, the later including numeric values of the animation.

4 Results of Ryan MUA Activity in SimCalc

The activity in classroom was done in pairs. Figure 3 shows one of the documents delivered; the description of the animation show that some algebraic procedures were done by hand in order to give numeric details of the motion.



Ryan comienza en x=1 con una velocidad positiva pero cada vez más lenta hasta tener una velocidad de 0 en el instante t=2.5 donde alcanza una posición de x=7.25, luego empieza a desacelerar cada vez más rápido hasta regresar al origen en el instante t=5.1925.

Fig. 3. A students' document with Ryan MUA activity.

Situated in SimCalc scenario, students visualize velocity graph intersecting the time axis, there velocity is zero, and changes from positive to negative values. Corresponding, there is the graphical event for a maximum in position graph. This is the kind of situated proof we have been managing for this calculus result. Class finishes but we ask students for homework to state the situation when velocity graph has initial value -5 and still intersects time axis in 5/2. No change is considered in Ryan's initial position. Homework is delivered by the platform.

Our pedagogical concern in this activity is to get to know the potential of SimCalc to promote the use of the different mathematical representations in order to have a complete description of the motion. It is possible that the visual image of position graph could work like a "meta-cognitive mirror" asking for the mathematical way to catch the time when Ryan crosses the origin (position equals 0 meters) of the straight line where motion takes place. This carries to the establishment of a quadratic equation and its solution, something to remember from students past experience with algebra.

We analyzed the information obtained; certainly we could not say that the activity is easy to all students, but the excitement we feel from them, makes us explore the way to improve the experience. We state the results organized in the Figure 4 where we tried to articulate the potential achievements when Ryan MUA SimCalc accompanies our thinking.

	Graphical representation		**Algebraic** representations (position and velocity)		
Total	Correct image with initial position 1	Incorrect image because initial position was also changed to -1	Correct notation and procedures	There are wrong procedures	Representations not given
24 students	19	5	12	5	7
100 percentage	79.17%	20.83%	50.00%	20.83%	29.17%

Interpretation of MUA motion			**Procedures** and explanations			**Excellent work**
Detailed and correct description	Correct description but incomplete	Description containing mistakes	There is a clear process of conversion between mathematical representations	There appear some inaccuracies with notations	No procedures, no explanations	**Excellent manage of mathematical representations**
12	6	6	10	5	9	6
50	25	25	41.67	20.83	37.50	25

Fig. 4. Results obtained during the activity.

5 Concluding remarks

Nowadays we can deal with the plotting of functions at a very early stage of the calculus' discourse. Technology software offers the opportunity for visual engagement and interaction. Through the experience we showed with SimCalc in classroom, calculus content seems better organized for students, offering powerful ideas to work with earlier. Motion situations over a straight line symbolized graphically allow students to feel confident with the way they communicate results, knowing what they are talking about when dealing with a function and its derivative.

We experienced how SimCalc could promote the learning through an initial global image that relates phenomenon of change of position in the motion context, and with the visual perception motivated by the interrelated graphs of velocity and position. In this scenario, co-action encourages the immersion to get to know how velocity behavior supports position behavior. This idea is important to understanding calculus as the mathematics of change and variation. That is why our concern since 2012 has been to deepen on it by using the PhD dissertation of one of the authors trying to control the desired events that technologies like SimCalc could bring to the learning of calculus.

We strongly believe that the graphical representation of function and its derivative could offer a new way to interpret globally the behavior of a magnitude that is changing. The relationship between the velocity-time graph and the position-time graph is part of a learning visual environment where the description of the motion is expressed through the behavior of those graphs.

Introducing SimCalc into the classroom has been a strong support to our approach for the learning of calculus, giving the opportunity to the reorganization of curriculum we have been working on. A new way to conceive graphs as common tools for mathematical reasoning brings into the classroom a powerful place for visual learning. Through the activity we discussed in this paper we showed how we can talk visually about maximum and minimum within a first meaning. In this way, the study of algebraic contents have meaning in order to access information about the motion and in order to plot functions, a goal that is desired at the end of a first calculus traditional course.

Now we can change the paradigm, we can deal with the graph of a function as the answer for predicting the behavior of a magnitude that is changing, and this since the beginning of the course. The graphical representation of functions is becoming a new way to think of and deal with change and variation phenomena, and it is the place where visual learners of this millennium will have a great opportunity to support their thoughts. Certainly, life in this millennium is an invitation to be visual.

Perhaps the notion of visualization brings the opportunity to identify the kind of cognitive action supported over a visual representation in order to extract information that could be expressed in numbers, in algebraic expressions or even in words. Visualization implies a consistent and organized reasoning involving all these elements and capable of producing new answers and also questions about the global event that stands in front of our eyes. Our proposal is to focus on graphs of both function and derivative, and promote cognitive actions in them, mathematical actions that will make numeric and algebraic information emerge. This is where our thoughts are looking for reaching a new educational technology paradigm, one that our new generations of learners deserve.

6 References

[1] Patricia Salinas, & Juan Antonio Alanís. “Hacia un nuevo paradigma en la enseñanza del Cálculo”; Revista de Latinoamericana de Investigación en Matemática Educativa, 12, 3, 355-382, 2009.

[2] Patricia Salinas, Juan Antonio Alanís & Ricardo Pulido. “Cálculo de una variable: Reconstrucción para el aprendizaje y la enseñanza”; DIDAC, 62-69, 2011.

[3] Patricia Salinas, Juan Antonio Alanís, Ricardo Pulido, Francisco Santos, Julio César Escobedo, & José Luis Garza. “Cálculo aplicado: Competencias matemáticas a través de contextos (Tomo 1)”. Cengage Learning, 2012.

[4] Lulu Healy. “TSG 15: The role and the use of technology in the teaching and learning of mathematics”; En Mogens Niss (Ed.), Proceedings of the 10th International Congress on Mathematical Education, 355-358, IMFUFA, Department of Science, Systems and Models, 2008.

[5] Abraham Arcavi. “The role of visual representations in the learning of mathematics”; Educational Studies in Mathematics, 52, 215-241, 2003.

[6] Raymond Duval. “A Cognitive Analysis of Problems of Comprehension in a Learning of Mathematics”; Educational Studies in Mathematics, 61, 103-131, 2006.

[7] Raymond Duval. “Un tema crucial en la educación matemática: La habilidad para cambiar el registro representación”; La Gaceta de la Real Sociedad Matemática Española, 9, 1, 143–168, 2006.

[8] Raymond Duval. “A crucial issue in mathematics education: The ability to change representation register. Regular Lecture”; In Mogens Niss (Ed.), Proceedings of the 10th International Congress on Mathematical Education, 1-17, IMFUFA, Department of Science, Systems and Models, 2008.

[9] Luis Moreno-Armella, Stephen Hegedus & James Kaput. “From static to dynamic mathematics: Historical and representational perspectives”; Educational Studies in Mathematics*, 68*, 99-111, 2008.

[10] Richard Noss & Celia Hoyles. “The technological presence: Shaping and shaped by learners”; Plenary Paper 10th International Congress on Mathematical Education, 2004.

[11] Luis Moreno-Armella, and Bharath Sriraman. “The articulation of symbol and mediation in mathematics education”; ZDM, 37, 6, 476–486, 2005.

[12] Luis Moreno-Armella & Stephen Hegedus. “Co-action with digital technologies”, ZDM, 41, 4, 505–519, 2009.

Embedded Programming for Computer Scientists

Peter M. Maurer
Dept. of Computer Science
Baylor University
Waco Texas, 76798-7356

Abstract – The past two decades have seen an explosion in embedded programming technology. Embedded controllers are everywhere, in practically every electronic device. These controllers, which are essentially computers, have become increasingly complex, with a corresponding increasing in the complexity of their programming. Embedded programming is a tremendous opportunity for computer science students. Here we give an overview of the three levels of embedded programming, bare machine, Real Time OS, and conventional OS. We also give suggestions for projects at each level. The aim is to allow the reader to construct his or her own course in embedded programming.

1 Introduction

Imagine yourself teaching Introduction to Computer Science. It's the first day and your classroom is filled with eager young students wanting to learn everything there is to know about computer programming. You give them their first "confidence builder" program, and they eagerly run off to the lab do the project (maybe!). However, the laboratory is not their first encounter with computers today. The alarm clock that woke them up for class has a computer in it, the microwave they used to warm their breakfast has a computer in it, the watch that told them when to leave for class has a computer in it, the cell phone that they (hopefully) silenced at the start of class has a computer in it, and if they drove to class, they interacted not with one computer, but with many. A car's electronic ignition has a computer, the automatic transmission has a computer, the anti-lock brakes have a computer, the intermittent windshield wipers have a computer, and on and on and on. If they drove a fairly new car, these computers did not work in isolation, but were consolidated into a communication network that controls the car as an integrated unit.

So the question is: who programs all of these computers? And who will program them in the future? Will it be your students? Will their undergraduate education prepare them to enter a world where everything they touch contains a computer? Will they understand even the basics of this type of programming? If yours is a typical undergraduate program, the answer is *probably not*.

There was a time when most embedded programming consisted of a few lines of code that could easily be hacked out by a non-expert. There are still embedded programs like this, but they are becoming the exception rather than the rule.

Processors designed for embedded programming are called "microcontrollers." A microcontroller is a computer in every sense of the word, except the name. As I am writing this, I'm looking at a sales blurb for a 32-bit microcontroller with more than half a megabyte of internal memory and a 32-megabyte external memory interface. (Cost: around $10.00 in quantity.) Any application that needs a device of this complexity also needs expert programming.

Of course, one can learn embedded programming "on the fly." One can learn anything "on the fly." It's usually better and faster to learn from a well-designed comprehensive course. Our purpose here is to discuss the design of such a course. There are, in fact, many textbooks devoted to embedded programming. Unfortunately, most of these focus on only one type of embedded programming. My aim here is to give a more broad-based view that will encompass most of the important issues. This will enable you to decide which topics are most important to you, so you can design your own course effectively.

The good news is that C has become the de-facto standard for embedded programming, and C++ is usually also available. Therefore the basic programming skills learned in an introductory programming course will also be used in embedded programming. (One can also use assembly language, but it's probably best to avoid this.)

There are three main levels of embedded programing. At Level 1, the programmer must write every line of code. There is no operating system and essentially no C library. The program is boot-strapped into the memory of a microcontroller, usually by burning it into non-volatile memory. Execution is initiated by a power-on interrupt. At Level 2, a miniature operating system (called a Real Time Operating System or RTOS) is used to assist with multi-tasking and other housekeeping chores. The RTOS, or portions of it, become part of the program and are boot-strapped into non-volatile memory along with the programmer's code. At Level 3, there is a real operating system, usually Linux. Special operations must be used to access the hardware features, but for the most part, you program the system as if it were an ordinary Linux system. The OS and the programmer's code are booted either from non-volatile memory or from a memory card.

It may be tempting to concentrate on Level 3, because the environment is familiar and extremely powerful. But this would give an inaccurate picture of the world of embedded programming. We recommend that all three levels be taught, with special emphasis on Level 1.

2 Before We Start

If you are not comfortable constructing simple hardware, or allowing your students to do so, it is best to cooperate with one or more colleagues from electrical or computer engineering. Even though we will focus on programming, the aim is to eventually use this programming to control special purpose hardware. It is best to have a number of pre-completed hardware projects available for students. These projects will have all the necessary hardware assembled and tested. There will be an empty socket that will eventually hold a microcontroller programmed with your students' code. At Level 1, five to seven projects will suffice. At Level 2, two or three complex projects will be needed. Simple projects at Level 2 can use the Level 1 hardware. At Level 3, one or two trivial projects should be used to familiarize students with the basic hardware, and then the sky's the limit.

If your students are comfortable with designing and building simple hardware, then they should be allowed to do so. The microcontrollers discussed here are available in DIP packages that can be used with standard breadboards. Hardware-capable students should be encouraged to branch out to more complex projects of their own devising.

3 Level 1: Bare-Device Programming

At Level 1 there is a small microcontroller, some simple hardware and virtually nothing else. The procedure for writing programs is quite different from that used for a typical console-level program. There is no I/O interface. The objects *cin* and *cout* don't exist. There is some debugging help available, but usually you will not be able to debug the program while it is running on the microcontroller.

First, you must decide which microcontroller to use. PIC [1] is the most popular, but we have found the AVR [2] 8-bit microcontrollers easier to work with. These are not the only choices (there are hundreds), but they are probably the two best options for a starting point. Another popular choice is the ARM [3] architecture because it is the architecture found in smart phones and tablets. However, the ARM architecture requires a substantial investment in development equipment.

For simplicity, we will describe the tools needed for AVR development. In our examples we will concentrate on the Atmel ATTiny85 microcontroller (75 cents in quantity), and will use the free development system provided by the ATMEL corporation (Atmel Studio). There are similar tools available for the PIC and other platforms.

The first concern will be memory management. There are three types of microcontroller memory, flash RAM, which holds the program, static RAM which is used for variables and stacks, and Electrically Erasable Programmable Read-Only Memory (EEPROM), which holds constant data. Although the EEPROM (and sometimes the flash RAM) can be written by the program, one needs to exercise care when doing so. Both EEPROM and flash RAM can endure only a limited number of write cycles before wearing out (typically 10,000 to 100,000 writes) so these types of memory must not be used for program variables. The static RAM can be written an unlimited number of times, but there is only a limited quantity available. Deep subroutine nesting will cause problems. Recursive programs are probably not a good idea. Large arrays probably won't work. The ATTtiny85, which we are using for an example, has 8 Kilobytes of flash RAM, 512 bytes of EEPROM, and 512 bytes of static RAM. The ATTtiny85 is, of course, rather small, but even the larger 8-bit microcontrollers have strictly limited amounts of memory.

3.1 Getting Started

To help you understand what embedded programs do, we'll start with a simple example. Consider the circuit of Figure 1.

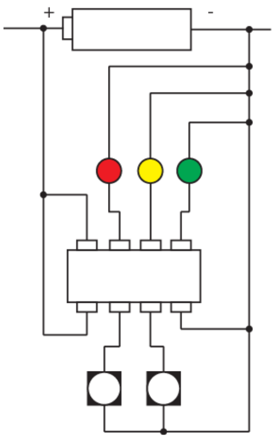


Figure 1. A Simple Circuit.

This circuit has an 8-pin, the TTiny85, three LEDs, and two buttons. Several programming assignments can be completed with this hardware. For a first assignment, we'll flash the LEDs on and off using the program of Figure 2.The TTiny85 has five general-purpose I/O pins, which are collectively called I/O PORTB. Each pin can be configured as an input or an output pin during program initialization. Programming is accomplished by moving values to memory mapped registers, which are defined as variables in the "avr/io.h" include file. These variables are DDRB, which is used to configure ports as input or output, PORTB, which is used to configure input pins and to set output pins to specific values, and PINB which is used to read the values of input pins. We won't use PINB in Figure 2. The Red, Yellow, and Green LEDs are attached to pins 2, 1, and 0, respectively. These pins are controlled by the three low-order bits of PORTB/DDRB/PINB. The left-hand button is attached to pin 3, and the right-hand button is attached to pin 4. These pins are controlled by bits 3 and 4 of PORTB/DDRB/PINB. The bits of each register are numbered from low order to high order starting with zero.

When setting the direction of a pin, a zero in DDRB indicates an input pin while a one indicates an output pin. For the input pins, it is necessary to use PORTB to set the internal pull-up resistors. This causes the default value of the pin to be 1. If we don't do this, the input pins will require extra external hardware to maintain their default values. A one-bit in PORTB activates the pull-up resistor, a zero lets the pin float.

```
#include <avr/io.h>
int main( ) {
    // initialization
    // Configure pins 0, 1, and 2 as output pins, 3, and 4 as input
    pins
    DDRB = 0x07;
    // activate the pull-up resistors of ports 3 and 4, setting the
    default input to 1
    PORTB = 0x18;
    // infinite loop
    for (;;) {
       PORTB = 0x1C; // turn on the RED LED
       PORTB = 0x18; // turn off the RED LED
       PORTB = 0x1A; // turn on the yellow LED
       PORTB = 0x18; // turn off the yellow LED
       PORTB = 0x19; // turn on the green LED
       PORTB = 0x18; // turn off the green LED } }
```

Figure 2. Rolling the Red/Yellow/Green LEDs.

Slowing the flashing down to the point where it is visible will be the student's first challenge. With the code of Figure 2, the flashing won't be visible. All three LEDs will glow dimly. To slow things down, we need to add a delay between turning the LEDs on and off. The simplest way to do this is to add a programmed delay as shown in Figure 3. (Remember that this is an 8-bit microprocessor, hence the *unsigned char* variables.) This delay might be enough, but it probably won't be. One can add nested loops to increase the delay to any desired value. Once the lights start blinking at the desired rate, we could try something fancier, like make the hardware work like a traffic light: *long red, short yellow, long green.*

The next challenge is to use the input pins to detect a button-push. With our sample hardware it will be necessary to debounce the button. A button does not go from off to on instantly. It goes from off, to rapidly changing between on and off, to on. When detecting a change in the button, the microcontroller program must wait a few milliseconds, and then confirm the status of the button. A change can be detected by reading a register and testing a bit. Debouncing can also be done with external hardware, which simplifies the programming. We will use the left-hand button is attached to pin 3. Figure 4 shows how to read the state of the button.

```
#include <avr/io.h>
int main( ) {
  unsigned char i,j,k;
    // initialization
    // Configure pins 0, 1, and 2 as output pins, 3, and 4 as input pins
    DDRB = 0x07;
    // activate the pull-up resistors of ports 3 and 4, setting the default input to 1
    PORTB = 0x18;
    // infinite loop
    for (;;) {
      PORTB = 0x1C; // turn on the RED LED
      for (i=255 ; i>0 ; i--);
      PORTB = 0x18; // turn off the RED LED
      for (i=255 ; i>0 ; i--);
      PORTB = 0x1A; // turn on the yellow LED
      for (i=255 ; i>0 ; i--);
      PORTB = 0x18; // turn off the yellow LED
      for (i=255 ; i>0 ; i--);
      PORTB = 0x19; // turn on the green LED
      for (i=255 ; i>0 ; i--);
      PORTB = 0x18; // turn off the green LED
      for (i=255 ; i>0 ; i--); } }
```

Figure 3. Visibly Rolling the Red/Yellow/Green LEDs.

```
#include <avr/io.h>
int main( ) {
  unsigned char i,j,k;
  // initialization
  // Configure pins 0, 1, and 2 as output pins, 3, and 4 as input pins
  DDRB = 0x07;
  // activate the pull-up resistors of ports 3 and 4, setting the default input to 1
  // Also, turn on the RED LED
  PORTB = 0x1C;
  // infinite loop
  for (;;) {
    // zero means the button is down
    if ((PINB&0x80) == 0 {
      // wait a bit
      for (i=255 ; i>0 ; i--);
      // is the button still down?
      if ((PINB&0x80) == 0) {
        // code to turn off the current LED
        // and turn on the next one is placed here
        while ((PINB&0x80) == 0) {
          // wait for the button to start back up  }
            for (i=255 ; i>0 ; i--); } } } }
```

Figure 4. Responding to a Button.

Completing these simple assignments will teach the student some important basic points. In embedded programming, “input” means reading the status of an input pin, “output” means setting the value of an output pin, and significant amount of code is required to configure the microcontroller for different operations.

As the preceding examples show, an embedded program consists of a set of initializations followed by an infinite loop. Delays of some sort are almost always necessary. Programmed delays are conceptually simple, but a better method is to place the microcontroller in a wait state, using an interrupt from an internal timer is then used to terminate the delay. This method consumes less power, because much of the microcontroller’s circuitry will enter a sleep state during the wait.

3.2 The programming process.

Microcontroller programs must be developed on a standard platform such as MS Windows and cross-compiled it to produce microcontroller object code. The object code is then downloaded into a microcontroller for execution. As mentioned above, we use the free software from Atmel Corporation for code development and cross-compilation. The object files are in Intel’s “.hex” format, which can be downloaded using a free software package called “avrdude” [4]. Figure 5 shows a picture of some sample hardware similar to that of Figure 1, but without the buttons. The black rectangle in the center is an 8-bit socket. The black rectangle below it is the microcontroller, which has been removed from its socket for programming.

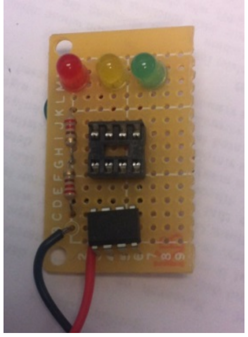


Figure 5. Sample Hardware.

Figure 6 shows an inexpensive programmer called the “Pocket AVR Programmer” [5]. It is about 1x1.5 inches in size and costs around $15. The gold socket at the top is a mini-USB connecter which is used to connect to a desktop PC or laptop. The cable at the bottom goes to a breadboard which is wired to communicate with several different microcontrollers. Figure 7 shows the breadboard with the microcontroller in place for programming. The connector to the programmer is at the far left.

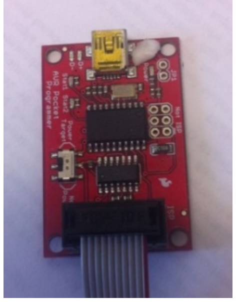


Figure 6. The Pocket AVR Programmer. [5]

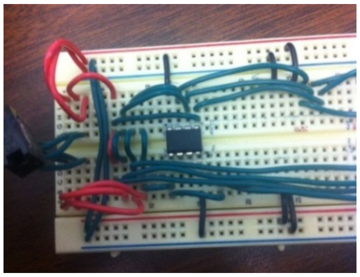


Figure 7. The Microcontroller in Place for Programming.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Once the microcontroller is programmed, it is placed back in its socket and the red and black wires at the lower left of Figure 5 are attached to a power supply. If one is very lucky, something will happen. However, in most cases, the program will have to be debugged before it starts to work properly. Many microcontrollers have in-place debugging features that can be invoked by a sophisticated development system, but with hardware shown in Figure 6, it is necessary to debug the program “offline.” The AVR software has an offline simulator that can be used to verify program correctness. It is best to simulate the program before downloading it for the first time. Once the program is working correctly in the simulator, it can be downloaded into the real hardware for further testing.

3.3 Advanced Programming

Additional assignments can explore the input and output features of the microcontroller as well as some of its internal features. The logical next step is to replace the programmed delays of Figures 3 and 4 with internal-timer based delays. This will result in an interrupt-driven program. This style of programming is substantially different from the examples given in Figures 2-4, but is necessary to access the more advanced features of the microcontroller.

Some of the additional features which should be explored are pin-change interrupts, analog input using pin comparators or analog-to-digital conversion. Analog-to-digital input is often used to determine the position of a dial attached to a potentiometer, but it could be used in connection with other features to record analog signals. Another important feature is analog output using Pulse Width Modulation (PWM). PWM can be used to vary the intensity of an LED, or control the speed of an electric motor. Some advanced microcontrollers provide digital-to-analog output, although this is rare.

Some microcontrollers provide Universal Serial Asynchronous Receiver Transmitters (USARTs) that can communicate with the serial ports of a conventional computer. This port can be read and written by a terminal program, allowing a user to communicate directly with the program on the microcontroller.

An increasingly important feature of embedded program is networking between two or more microcontrollers using either the Serial Peripheral Interface (SPI) protocol or the Two Wire Interface (TWI) protocol. These protocols can be explicitly programmed, but most microcontrollers provide built-in hardware which simplifies the programming process.

Another important consideration is power management. Certain features permit unused portions of the microcontroller to be powered down, and others permit the microcontroller to enter various levels of sleep modes when waiting for events to occur. These modes of operation are extremely important for battery-powered circuits, so it will be beneficial for students to learn about them.

4 Level 2: Real-Time Operating Systems

At level 2 we add an operating system, the main benefit of which is multi-tasking. Everything we learned at level 1 is still valid. We do things in much the same way as at level 1, but now we can control multiple independent devices simultaneously with only moderate difficulty. There are many microcontroller-based Real Time Operating Systems (RTOS) available, some of which are free. One of the most popular is FreeRTOS [6], which is available for a number of different platforms, including the AVR ATMega series. Unfortunately, the larger RTOS’s are too large for the smaller microcontrollers such as the ATTiny85. An even smaller RTOS, FemtoOS [7], is available for these smaller microcontrollers.

The OS itself is the only new tool needed at this level. Cross-compiling, simulating, and downloading, are done using precisely the same tools used in Level 1.

The first step in creating an RTOS-based program is to copy the OS files into your project. The OS itself becomes the project with your code consisting of a set of subroutines that are called by the OS at various times. The main routine will be provided by the OS, but there will be a primary subroutine for each task. Each task will consist of an initialization section followed by an infinite loop. Tasks are pre-emptible via interrupts, and will be scheduled according to a system of priorities which you can designate using configuration files. Operating system calls are available for task synchronization, timed waits, and inter-task communication.

It will generally be necessary to modify the OS source files to some degree, although these modifications will normally be confined to the OS configuration files. Because of this it is usually a good idea to start with a sample project that is as close as possible to the project you wish to complete. You can then modify the sample project to do what you want. This saves the trouble of creating configuration files from scratch.

Figure 8 shows a two-task FemtoOS project using the ATTMega168 microcontroller. Two pins of the microcontroller are wired to LEDs, one red and one green. These two pins are controlled by the two least significant bits of the devLedPORT and devSwitchPIN registers. The variables devLedPort, and devSwitchPIN, are mapped to microcontroller registers by the OS include file. The button is wired to a pin controlled by the third bit of the devSwitchPIN variable, which is also mapped to a device register.

The program blinks the two LEDs alternatively, switching from one the other when the button is pushed. There are two tasks, one to manage the blinking, and another to watch the button. A global variable is used for inter-task communication. The programming of device registers is essentially the same as at level 1. However, delays are handled through the OS, not by programmed loops.

The two functions “appLoop_LEDtask1” and “appLoop_DoButton” are the main routines for the LED task and the Button task respectively. The “TimeToSwitch” variable is used for task communication. The first loop in the LED task turns on the red LED, delays 200 milliseconds, then turns off the red LED and delays another 200 milliseconds. This procedure is repeated until the “TimeToSwitch” variable is set to true by the Button task. When “TimeToSwitch” becomes true, the first loop is broken, “TimeToSwitch” is set back to false, and the second loop is entered. The second loop repeatedly turns on the green LED, delays 200 milliseconds, turns off the green LED, and delays another 200 milliseconds. As with the first loop, if the button task sets “TimeToSwitch” to true, the loop is broken, “TimeToSwitch” is set to false, and the task returns to the first loop.

The Button task waits for a button to be pressed. When a button press is recognized, the “TimeToSwitch” variable will be set to true. The LED task has the responsibility for setting “TimeToSwitch” back to false. The Button task is a bit more complicated than one might expect, because it must debounce the button. The Button task waits for the button to be pressed. This will be detected as a change in bit three of the devSwitchPIN register. If such a change is detected, the task delays 100 milliseconds, confirms that the bit has indeed changed, and sets “TimeToSwitch” to true. Then the task waits for the button to be released. This will cause bit three of devSwitchPIN to return to zero. Waiting is done by repeatedly delaying 100 milliseconds and retesting the devSwitchPIN bit. Finally, the task waits for “TimeToSwitch” to be set to false by repeatedly delaying for 100 milliseconds. After “TimeToSwitch” becomes false, the Button task resumes waiting for the next button press.

The code of Figure 8 is compiled and simulated using the same tools as were used in Level 1. The initial include statement inserts the required OS definitions into the users program. The FemtoOS source-code files must be made part of the project, which allows the OS code to be compiled into the user's program. Once the code has been simulated, the object file is downloaded to the microcontroller and run just as in Level 1.

```
#include "femtoos_code.h"
static Tbool TimeToSwitch = false;

void appLoop_LEDtask1(void) {
  for (;;) {
    for (;;) {
      devLedPORT |= 1;
      taskDelayFromNow(200U);
      devLedPORT &= 0xFE;
      taskDelayFromNow(200U);
      if (TimeToSwitch) break;   }
    TimeToSwitch = false;
    for (;;) {
      devLedPORT |= 2;
      taskDelayFromNow(200U);
      devLedPORT &= 0xFD;
      taskDelayFromNow(200U);
      if (TimeToSwitch) break;   }
    TimeToSwitch = false; } }

void appLoop_DoButton(void) {
  for (;;) {
    if (devSwitchPIN & 4) {
      taskDelayFromNow(100);
      if (devSwitchPIN & 4) {
        TimeToSwitch = true;
        while (devSwitchPIN & 4) {
          taskDelayFromNow(100);  }
        while (TimeToSwitch) {
          taskDelayFromNow(100);  } } } }}
```

Figure 8. Switch from One Blinking LED to Another.

The programming exercises for Level 2 should be more complex than those for Level 1. To facilitate this we recommend that a larger (in terms of pin count) microcontroller be used, primarily because it is easier to control multiple devices if more pins are available. We suggest the ATTiny861, or the ATMega168.

5 Level 3: A Complete System

Level 3 is quite similar to ordinary Linux programming. There are many devices available, but we recommend the Raspberry PI [8] which is available for less than $50. It is a complete Linux system that can be used as a desktop computer, if you wish. (As a desk top, it is rather slow.) You can use it both as a development environment and as part of an embedded system. There are commercial kits available for the Raspberry PI, that cover everything from robotics to home theater. There are even educational laboratories aimed at the K through 12 audience. There is online documentation (available from many sources) that covers programming, OS modifications, and sample projects.

It is best to ignore all the commercial products and focus on programming the general purpose I/O pins. This is done in the same manner as in Level 1, namely by moving values to or reading values from memory mapped registers. Figure 9 shows a picture of the Raspberry PI. The General Purpose I/O (GPIO) pins are at the lower right. The device is booted from an SD card, which is the rectangle at the far right. The device pictured is wired to operate as a desktop computer, with the video port at the bottom, power at the upper right, internet connector at the upper left, and USB hub at the center left. Mouse and keyboard are attached to the USB hub.

General purpose I/O is managed through a Broadcom BCM2835 ARM peripherals chip [9], whose GPIO registers are mapped to the Raspberry PI's memory space starting at 0x2002000. Figure 10 shows the general outline of an LED flasher task. Unlike the ATTiny and ATMega microcontrollers, the Raspberry Pi has two separate registers, one for setting pins to 1 and another for setting pins to 0. (Such variations between different products are common.)

The LED must be connected between pins 7 and 8, with a resistor to limit the voltage. Figure 11 shows the pinout of the GPIO pins. In Figure 9, Pin 1 is the rightmost pin of the top row. The pins marked as GPIO can be used for any purpose and can be configured as input or output. The other pins (other than power and ground pins) can also be configured for general purpose I/O if their specialized functions are not required.

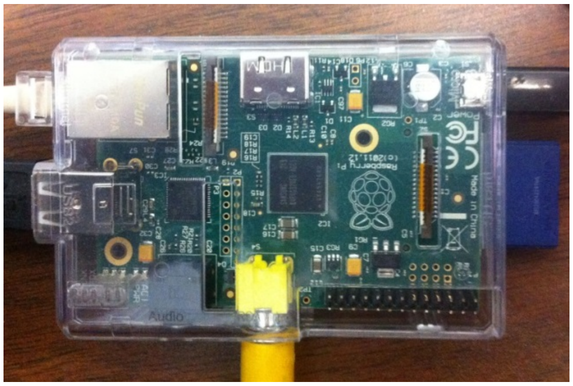
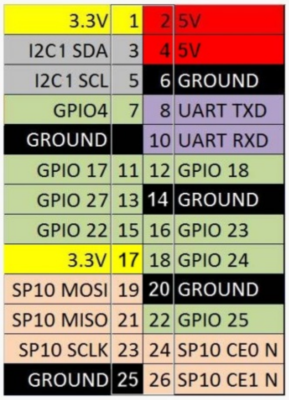


Figure 9. The Raspberry PI.

```
Configure Pin-7 as an output, using Function-Select   register.
for (;;) {
    Move 1 to Pin-7 GPIO-Set register
    Pause
    Move 1 to Pin-7 GPIO-Clear register
    Pause }
```

Figure 10. A Raspberry PI LED Flasher Program.



3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND		10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

Figure 11. Raspberry PI GPIO Pinout [6].

The possibilities for the Raspberry PI are nearly limitless. The SP10 pins (See Figure 11) can be used to communicate with Level 1 or 2 microcontrollers using the SPI protocol. The I2C1 pins do the same using the TWI protocol. This gives one the power to construct complex systems with Level 1 or 2 microcontrollers performing the low-level tasks and Linux-based programs providing the over-all control.

The UART pins can be used to communicate with a serial port on another computer, or with a microcontroller USART interface. There are several other connectors in addition to the pins described in Figure 11. There is a camera connector near end of the Ethernet connector. There is an LED screen connector above the memory card slot. There is a JTAG interface and an additional GPIO header. (These last two have no pins attached and require expert handling.)

The power of this $30 device is astounding. I will leave the exploitation of this power to your imagination.

6 Conclusion

As mentioned above, the past two decades have seen an explosion in embedded programming. Microcontrollers have proliferated into almost every electronic device. In addition, controllers have become increasingly more complex and increasingly less expensive. The explosion in embedded programming can only be expected to continue. As systems become more complex, there will be an increasing need for expert programmers to program them. This is an important opportunity for computer science students, one that cannot be ignored. A one-semester course in embedded programming will prepare students for this exciting and rewarding opportunity.

7 References

1. PIC Microcontrollers http://www.microchip.com
2. AVR Microcontrollers http://www.atmel.com
3. ARM Cortex Microcontrollers http://www.ti.com/lsds/ti/arm/overview.page
4. AvrDude http://www.nongnu.org/avrdude
5. Pocket AVR Programmer http://www.sparkfun.com
6. FreeRTOS http://www.freertos.org
7. FEMTOOS http://www.femtoos.org
8. Raspberry PI http://www.raspberrypi.org
9. BCM2835 http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf

Running an Agile Class

Kevin A. Gary
Arizona State University
7171 E. Sonoran Arroyo Mall
Mesa, AZ 85212
kgary@asu.edu

Abstract

Agile methods, particularly Scrum, apply an empirical process model to the complexity of software development. The reasoning, in short, is that a highly iterative process with specific short-term goals can be instrumented to provide constant feedback to change and disruption. Agile methods focus on short iterations, adaptive work assignments, constant feedback, and process visibility to address the fundamental nature of complexity. The author is currently experimenting with agility in the classroom through the incorporation of these mechanisms and the support of online tools. Online tools can help in several ways, such as helping the instructor and teaching assistants scale up the agile teaching process. Initially we have found that the most impactful of online tools is a scrum taskboard, or *Scrumboard*, to make the work of the class visible to all stakeholders. In the context of the technology-supported classroom, the online Scrumboard not only helps the instructor and teaching assistants (yes, the *pigs*) manage the busy work of running the course, but lets the stakeholders (students, or *chickens*) understand what is happening now and next, and how their progress influences what happens on the board. This paper will focus on the use of an online Scrumboard in an upper-division project course, sharing the instructor's experience combined with a survey completed by the students. It will expand on agile teaching and its potential for managing today's complex higher education class.

Introduction

This is **not** another paper describing how to teach agile methods to software engineering students. Rather, it is a paper on experiences using an agile methodology to execute a college course. This paper shares instructor experiences and student evaluations of running a course as a series of scrum *sprints*, and sharing the visibility of the sprints via an online *scrumboard*.

The higher education landscape is changing in many ways; online courses, project-based learning, active/discovery-based learning, technology aids, high articulation rates, and so on. The "sage on the stage" teaching model is as obsolescent as the waterfall model in software engineering. Instructors are now expected to balance a fast-paced classroom incorporating new learning models and complex course plans with the events that inevitably arise when dealing with nonlinear processes. In other words, dealing with the complexity of teaching material while guiding self-directed activities to teams of students with different educational backgrounds.

The experience described in this paper is from an upper-division project-centric course where student teams work on a project while learning and applying new concepts. The author found that constantly revising a course plan based on student progress was a losing game; there were too many events that altered the plan. The "instructor as coach" model was difficult to scale, students were confused between the learning process and the project process, and too much time was spent tweaking the course *with the students* than was spent on executing the learning process.

The author's epiphany was realizing an inability to control a predictive (course) plan and instead applying a reactive, empirical process model. Agile methods, notably Scrum, provide mechanisms for implementing empirical process control. *Sprints*, *sprint goals*, and *continuous feedback* provide for directed process execution with the ability to adapt to change [6]. In terms of a course, a sprint is a unit of time; a sprint goal is the outcome (or suboutcomes) to be achieved by the sprint, and the class meeting time, formative and summative assessment activities, and technology-based touchpoints (email, project dashboards, discussion forums) a form of continuous feedback. This paper describes the agile mechanisms put in place and shares the results of a small survey of class participants on the utility of the approach. Some thoughts on progressing the idea of agile classrooms are provided at the conclusion of the paper.

The Benefits of Agility in a Course Context

Agile methods are, of course, the silver bullet software engineering has long awaited. It provides productivity, innovation, creativity, time-to-market, under budget costs and zero learning curve to all adopters on their software projects. Plus, it cures the common cold. So why shouldn't we shoot that bullet at the difficulties in the changing landscape of higher education? After all, the modern classroom is distancing itself day by day from the classrooms of yore. Learners arrived with predetermined expectations of what energy they should and will expend on the class. Technology is as much an intrusion as an innovation; we swing the Mjölnir wildly with tablets, cell phones, content delivery systems, online interactive collaboration and of course MOOCs (my how the mighty have fallen…). New pedagogies exuberantly attack the sage on the stage with enough variants to start their own revolution (oh, wait…).

Facetiousness aside, Agile is in the category of hyped technologies and some are starting to track its spiral into the "Trough of Disillusionment" [4,7]. Most critics do not fault agile methods themselves, but rather the inexperienced or inaccurate application of agile methods by uninformed middle management. Of course over time agile methods will mature into better-understood variants and adaptations and appears to be here for the long haul [5]. It is the author's experience that Agile experienced an initial popularity as part of a developer revolt of sorts; emboldened by the success of open source software and community development spurred by the Internet, developers rapidly adopted (sometimes in secret) agile means of executing their tasks. Traditional project and product managers did not always buy into these methods, as they did not understand them, and felt more comfortable with traditional project management techniques and tools, and needed more long-term guarantees than agilist developers would provide (ignoring the fact that these guarantees only guaranteed failure).

Much of the agile literature espouses the wonders of agile from the software engineering perspective, but there are also several business-oriented benefits that have helped agile overcome initial project/product management skepticism [3]. Most notable of these are an ability to adapt to changing requirements, expectations, and operating conditions, an ability to cross functional boundaries, and the transparency and visibility of the process to all stakeholders. While there are many other benefits of agile, it is these benefits that convinced the author that perhaps *agile* applied to *teaching and learning* could address some of the issues arising from the evolving education landscape. Specifically:

- The changing environment in which the modern university course is taught is changing as mentioned above. Changes in student expectations, changes technology, changes in pedagogy map to the changes in requirements and operating conditions that agile addresses for software development organizations. Can agile help manage this change for the educator?
- More and more (at least at the author's institution), higher education students are encouraged to work in multidisciplinary teams and even define custom multidisciplinary degree programs. At the author's institution students may cluster 4 courses into a *secondary focus area* combining a student's interests with degree program outcome requirements. The result from the faculty perspective is more students showing up in upper division classrooms with a variety of backgrounds and interests. This maps to the agile benefit of dealing with cross-functional boundaries.
- Transparency and visibility of the process is a tremendous impetus for agile adoption on software development projects. Project managers enjoy being able to receive constant feedback on projects via daily standup meetings and dashboards, while product managers and clients gain visibility and control into the management process via short iterations and frequent planning sessions. Risks are identified early and addressed through a scale-down / scale-up set of mechanisms. To the author this is the main potential benefit of applying an agile approach to teaching and learning. Providing students with visibility into where the course is, what learning goals are being achieved (in the current iteration), and getting constant feedback provides them a context and organization of the learning process, resulting a less frustrated and more engaged student participant.

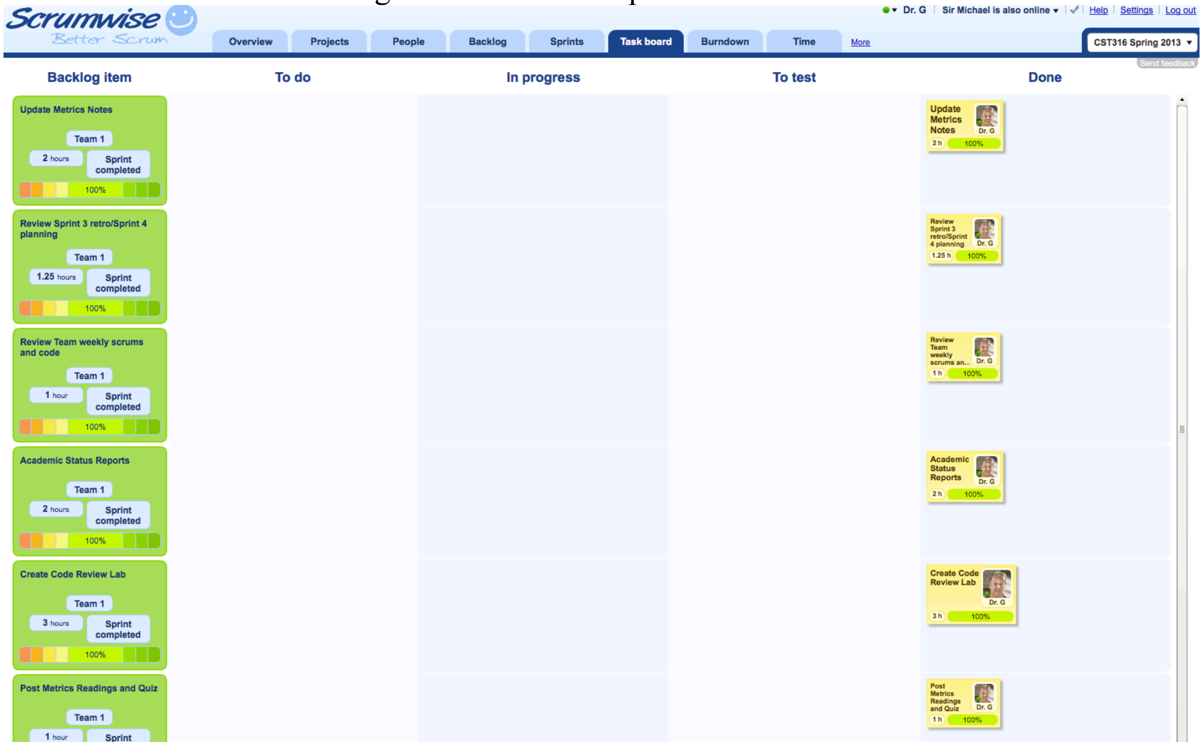
The next section will present the specific agile mechanisms put in place in an upper division project-centric course in the spring of 2013, with the focus on achieving the third agile benefit of transparency and visibility.

Agile Practices for Course Management

The traditional means by which a higher education instructor manages her/his course offering is to distribute a planning document, or *syllabus*, on the first day of class. This syllabus is usually an example of a prescriptive process model – it contains process steps (course topics and perhaps a calendar), process constraints and resources (textbooks, homework submission policies), process assessment (learning outcomes and grades), and policies for resolving process errors (grade appeals, academic integrity policies, etc.). While usually prescriptive there is nothing mandating that it be constructed as such; one can argue that the most important content of the syllabus are the process assessment components, as they inform the student what s/he will learn and how feedback will be provided to ensure the process of learning in the classroom supports achievement of those outcomes. Nonetheless most courses map out a syllabus and use it as a plan to prescriptively execute throughout the semester. Technology only reinforces this approach, as many course management systems now provide a syllabus tool and allow the instructor to generate learning activities from that tool, and often map them onto a course calendar.

But what if the instructor focused less on course content and learning activities and more on learning outcomes, continuous feedback, and making in-time adjustments to learning process execution during the semester? Then s/he starts to sound agile. To realize this approach, the author introduced some agile mechanisms based on the Scrum methodology [6] in a junior level project-centric course in the spring semester of 2013. The course material focused on software development best practices, though that is not particularly important outside of there being specific measurable technical competencies as part of the course outcomes. 23 students were enrolled in the course, 22 of the 23 passed the course.

The principal agile concept introduced was the Scrum *sprint* with accompanying *product* and *sprint backlogs*. Using the Scrumwise tool, obtained via a free academic license, the instructor created a product backlog by starting from the course learning outcomes. From there, eight sprints were mapped out over the course of the semester corresponding to eight modules the instructor wished to cover. Figure 1 shows a completed Scrumboard from Scrumwise.



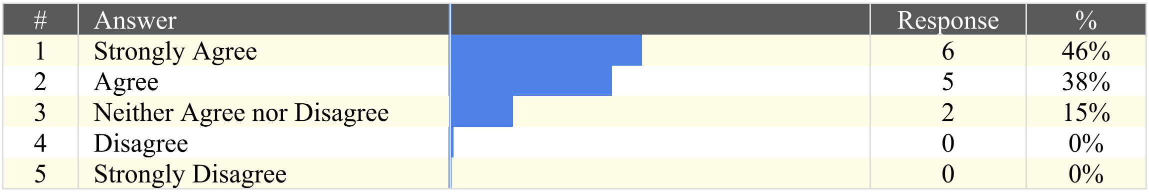
Backlog item	To do	In progress	To test	Done
Update Metrics Notes Team 1 2 hours Sprint completed 100%				Update Metrics Notes Dr. G 2 h 100%
Review Sprint 3 retro/Sprint 4 planning Team 1 1.25 hours Sprint completed 100%				Review Sprint 3 retro/Sprint 4 planning Dr. G 1.25 h 100%
Review Team weekly scrums and code Team 1 1 hour Sprint completed 100%				Review Team weekly scrums an... Dr. G 1 h 100%
Academic Status Reports Team 1 2 hours Sprint completed 100%				Academic Status Reports Dr. G 2 h 100%
Create Code Review Lab Team 1 3 hours Sprint completed 100%				Create Code Review Lab Dr. G 3 h 100%
Post Metrics Readings and Quiz Team 1 1 hour Sprint				Post Metrics Readings and Quiz Dr. G 1 h 100%

Figure 1. Completed Scrumboard from a course sprint

The notion of a sprint is fundamental to the introduction of agile teaching and learning. First, the sprint provided a time-boxed way to achieve a learning outcome within a course, or at least to cover some material associated with a learning outcome. Second, the concept of a *sprint goal* is very important to this process. A sprint goal in Scrum is the specification of what the team is trying to achieve by completing the sprint. It is best used as a litmus test to determine whether the sprint was successful or not – instead of assuming a sprint is successful merely by completing all the tasks associated with a backlog item. This is an important, oft-overlooked aspect of Scrum, and one that enables it to react to change so well. The focus of a sprint is on achieving a goal, not on completing activities. Likewise, for a teaching and learning process, the focus should be on understanding whether students are achieving learning outcomes and not as much on whether they actually (or the instructor actually) completes various tasks. Finally, the Scrumboard provides a powerful *information radiator* [2] for the class. It provides constant feedback as to what the instructor is doing, what is coming next in the class, what is changing, and what the goal is.

The Scrumboard for the course was made available throughout the course by adding the students as stakeholders ("chickens" in Scrum lingo) while the instructor and teaching assistants were the owners ("pigs"). Anecdotally, the author found that utilizing the Scrumboard and organizing course tasks in sprints was a great way to manage the constant change and minutiae of executing a course. Interestingly, sharing the Scrumboard with students appeared to be a step toward better communication of weekly course goals and activities, resulting in less communication time around where the class was and what was coming next [1]. Of course this is the perspective of a single instructor. A survey was also conducted asking the students for their perspectives on utilizing Scrum in the class. The survey asked 3 simple rating questions and one qualitative free-form type of question. Of the 23 enrollees, 13 responded to the survey.

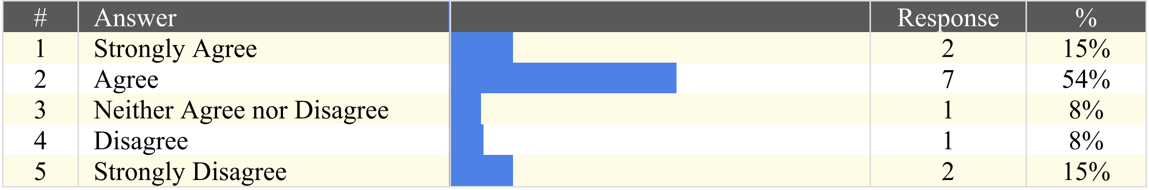
Q1: "I liked being able to see what the instructor was preparing next for our XXX class"



#	Answer		Response	%
1	Strongly Agree		6	46%
2	Agree		5	38%
3	Neither Agree nor Disagree		2	15%
4	Disagree		0	0%
5	Strongly Disagree		0	0%

Table 1. Results from survey question Q1

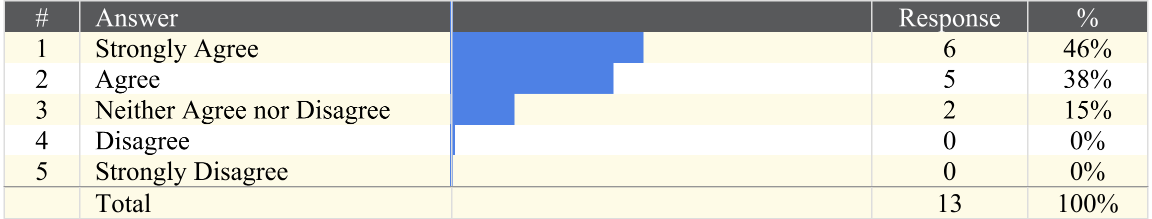
Q2: I checked the XXX Scrumboard periodically throughout the semester



#	Answer		Response	%
1	Strongly Agree		2	15%
2	Agree		7	54%
3	Neither Agree nor Disagree		1	8%
4	Disagree		1	8%
5	Strongly Disagree		2	15%

Table 2. Results from survey question Q2

Q3: I would recommend the Scrumboard for the next offering of XXX



#	Answer		Response	%
1	Strongly Agree		6	46%
2	Agree		5	38%
3	Neither Agree nor Disagree		2	15%
4	Disagree		0	0%
5	Strongly Disagree		0	0%
	Total		13	100%

Table 2. Results from survey question Q3

(Note: XXX is used to hide the course identification)

Q1 directly asks whether students found the Scrumboard useful, and most students responded positively. Likewise Q3 is similarly positive, though a cross-tabulation of the responses reveals that respondents did not answer these 2 questions the same way, despite the identical aggregate results. Q2 responses were more distributed, though 9 of 13 answered positively while only 3 of 13 answered negatively. The cross-tabulation reveals that for those 3 that answered negatively to Q2 (i.e. they did not check the Scrumboard periodically), 2 of those 3 answered Q1 and Q3 positively. The author believes this is because a postmortem reviews of the class Scrumboard at the end of the semester led to observations that some students did not track the board closely as they did not fully understand the intent of these sprints in the first place.

Q4: Please enter any thoughts or recommendations you would make regarding your experience using a Scrumboard by the Instructor in XXX

There were only 7 responses to this question, and most were comments on the Scrum tool employed and not on the Scrumboard itself. That is, as often happens with open-ended class style evaluation surveys, the students riffed on what they wanted, not what was asked! One exception:

> *"It provided a good reference for where we were in the class and what to expect. But I did not use it as much as I probably should have."*

Which the author believes represents the typical perspective of students in the class.

These results are taken from a single semester offering with a relatively small number of participants in a single course section, so the author is aware of the limitations of drawing conclusions from the survey. Further the survey asks about perspectives on learning and the utility of a teaching and learning tool, and does not directly measure learning outcomes. Future iterations require a measurement vehicle where the impact of the methodology and tools on learning outcomes is directly addressed.

Discussion and Future Directions

This is a new way of running a class, and communicating the model and the mode of work to the stakeholders (the students) is critical to its success. Confounding this challenge was the fact that students were also executing their own sprints for their class projects. While at first the author thought this would help minimize the learning curve for conducting an agile class, it appears to have confused the matter for some students (as evidenced by Q2).

The careful reader has probably noticed by now that the definition and execution of the sprints does not exactly match the motivational philosophy given for employing sprints above. Most of this is due to the simple nature of trying this in a real class the first time, and the practical obstacles that arise. For example, eight sprints were defined because there were eight pre-existing curricular modules available for the course based on previous course offerings. It was thought that a straight one-to-one mapping of modules to sprints would be the least intrusive way to introduce agility. But, the results were sprint goals narrowly defined by technical competencies (e.g. "the student will be able to construct a unit test in tool X") and not easily mapped to course level outcomes (e.g. "at the conclusion of this course, students will understand the principles of quality source code development, the challenges in introducing best practices on software development projects, and apply tools and techniques for achieving code quality in the context of a scalable project"). The author had to create a suboutcome layer and map these suboutcomes, representing specific technical competencies to the course outcomes, and all this was tracked outside the sprints. For the next iteration of the course in Spring 2014, the author is considering how to introduce *epics* as a means to manage *themes* that corresponding to learning outcomes. For example, one epic might represent quality coding practices with sprints dedicated to static analysis, refactoring, and code reviews (3 example modules from the course) while another epic might represent software testing and include sprints on unit testing and continuous integration and testing (2 more modules). Other approaches may work as well.

Despite the infancy of this research, some clear benefits have been identified. Transparency, visibility, and organization are powerful vehicles for progressing learner motivation and outcomes. This is inline with larger studies on these basic mechanisms (a good summary recently published here [1]). The nature of change may be distinct from software engineering, but the transformation of the modern higher education experience means that change does occur and takes many forms, and perhaps *empirical process control* is one mechanism to help the overwhelmed instructor adapt. Of course not all is good; there is danger is adopting a process model from one domain to another domain of practice, though we note ironically this is what happened with Scrum in the first place. Many of the drawbacks encountered so far are likely due to the infancy of the work combined with this danger. Figuring out the right agile practices, defining a methodology for introducing those practices in the classroom, and assessing their impact requires deeper investigation and rigor, and this is the trajectory of future work.

Finally, the author notes that this work is part of a larger research impetus to investigate and implement strategies taken from engineering and computing disciplines in the context of teaching and learning processes. Software engineering in particular has seen great change in the understanding of best practices, driven by new process models like agile but also from the expansion of community-oriented development spurred by the Internet. In short, software engineers are becoming expert practitioners in working on problems with high rates of change – in requirements, methods, tools, technology, and expectations. That sounds a lot like what is starting to happen in higher education, so perhaps we should look to process and practices we have right under our nose for the solution!

References

1. Barrett, D., "Teaching Clearly Can Be a Deceptively Simple Way to Improve Learning", *The Chronicle of Higher Education*, November 22, 2013.
2. Cockburn, A. (Online), "Information Radiator", Available at http://alistair.cockburn.us/Information+radiator, (last accessed January 4, 2014) June 19, 2008.
3. Highsmith, J. and Coburn, A., "Agile Software Development: The Business of Innovation", *IEEE Computer* September 2001, pp. 120-122.
4. Janes, A., Succi, G., "The Dark Side of Agile Software Development", Proceedings of the Third ACM conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH 2012), Tucson, AZ. October 2012.
5. Kajko-Mattsson, M., Aguiar, A., Boness, K., Kaindl, H., Pooley, R., and Tael, A., "Long-Term Perspective of Agile Methods", Panel report in the Proceedings of the 4th International Conference on Software Engineering Advances, 2009 (ICSEA '09), Porto Portugal, September 2009.
6. Schwaber, K. and Beedle, M., *Agile Software Development with Scrum*, Prentice-Hall 2001.
7. Wilson, N. (Online), "The Trough of Disillusionment" Available at http://blogs.gartner.com/nathan-wilson/the-trough-of-disillusionment/, (last accessed January 4, 2014) July 12, 2012.

How can Augmented Reality favor the learning of Calculus?

P. Salinas[1], **E. Quintero**[2], **and E. González-Mendívil**[2]
[1]Mathematics, ITESM, Monterrey, Nuevo Leon, Mexico
[2]Mathematics, ITESM, Monterrey, Nuevo Leon, Mexico

Abstract - *This paper presents the use of augmented reality for educational purposes within a grounded innovation process in college for the calculus curriculum. The aim of the application developed is offering students an opportunity to deal with spatial visualization skills. An activity performed in spring semester 2014 in a Calculus I course with 20 students shows the positive impact regarding the interaction with mathematics in this mode. The AR Application from 2D to 3D was provided to students in tablets prepared before with the whole activity. As part of the data collection involving the multiple source of information that a case study includes, a brief survey with open-ended questions was applied. We share the descriptive analysis of its results which make us believe that the abstraction feature of mathematics feels reachable when a virtual object stands in front of our own space and time reality.*

Keywords: Augmented reality, Calculus, learning, innovation.

1 Background

We work at Monterrey Institute of Technology and Higher Education, a private educational institution in México whose educational model cares for the integration of technology in the learning process. Our particular concern is Mathematics education, and as part of an innovation program, we have been combining the use of emerging technologies and educational methods in order to achieve a better math learning. Today we identify ourselves as TEAM, *Tecnología Educativa para el Aprendizaje de las Matemáticas*. We took on the challenge of creating a multi-discipline team that could lead an innovation process for the learning of Calculus in our Campus Monterrey.

Nowadays technology brings the opportunity to transform mathematics curriculum. The way we interact with math knowledge is changing. The algebraic paradigm is not the only way to present Mathematics anymore. Today we can admit that the visual aspects of Mathematics are available everywhere and within our reach, just as our mobile devices are. Why not thinking about a classroom with plenty of images showing the beauty of mathematics knowledge?

Technology changes the way we learn, and students in this millennium are already prepared to live learning experiences through visual and gestural interaction. As part of TEAM we are confident of the advantages in this new paradigm under construction. We are also conscious of the challenge we have in order to make the most of technology with an educational goal inside the higher education institutions. We must be aware of new technology development and, as a matter of fact, we truly enjoy knowing about it. We seek to transform the learning of mathematics through an enjoyable interaction with technology.

The experience we have about Mathematics Education includes the educational research that allows a grounded innovation process in the calculus discourse for college. Three PhD dissertations in Mathematics Education and several Master degrees have been dedicated to build a curriculum innovation in our institution. The new approach promotes the transfer of calculus knowledge to the different specialty branches for engineering, favoring the student's understanding of the application of this knowledge solving real life problems [1, 2 & 3].

The practice with the new didactic for the teaching and learning of Calculus at engineering programs is possible by the development of a training program with teachers at Campus Monterrey. Teachers live the experience with the authors about the way to deal with the textbooks and electronic resources putting at their service. Discussions and reflections about the educational process invite them to make use of technology in classroom. The textbooks collection Applied calculus; mathematical competencies through contexts I, II, and III offer students and teachers a new structure to be introduced to calculus notions and procedures [4, 5 & 6].

Even when this redesign of calculus discourse integrates technology, like spreadsheets and specialized graphical software, now as TEAM we inquire about the introduction of the new emergent technologies. We want to focus learning on the development of certain skills that have not been explicit in the curriculum but nevertheless are important when you deal with mathematics. We believe that in the teaching of calculus some cognitive skills have been taken for granted. Some of them result basic for the understanding of mathematics and useful in problem solving processes. Spatial visualization, for example, has been assumed as an innate skill in the students. Experience with the teaching of solids of revolution may question if this is so.

We could identify spatial visualization as a cross-curriculum content and took on the task of designing an educational resource improving its development. Augmented Reality technology has been our choice to begin our research about its relative advantages.

2 An Augmented Reality Application

Last year we presented at VARE 2013 (Virtual and Augmented Reality in Education International Conference) a prototype of Augmented Reality educational resource. We worked on the design to serve the purpose of supporting the development of spatial visualization skill in students of Calculus I, II and III at college level. The prototype included the part corresponding Calculus I. We are, at the time, working on Calculus II and III.

For VARE presentation we included a case study where 30 students were invited to get to know the application outside the classroom. They didn't know how to use it or how it was done. The experience allows us to understand the students' attitude when interacting with the technology and the devices it included. It also gave us information about certain improvements already done now [7].

In general terms, we found the product acceptance and positive impact on students regarding interaction with mathematics in this mode. We were looking forward to continue with the remaining two parts of the product. Now TEAM moves to complete the Application with the sponsor of Novus Calls 2013 and 2014 that our institution offers in order to promote innovation projects. Figure 1 and 2, respectively, show the graphical user interfaces for the cover and the thematic content of the app.

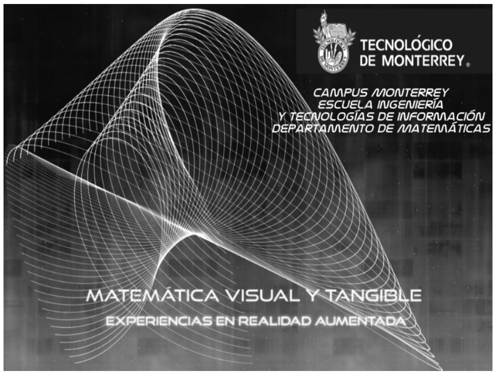

TECNOLÓGICO
DE MONTERREY.
CAMPUS MONTERREY
ESCUELA INGENIERÍA
Y TECNOLOGÍAS DE INFORMACIÓN
DEPARTAMENTO DE MATEMÁTICAS
MATEMÁTICA VISUAL Y TANGIBLE
EXPERIENCIAS EN REALIDAD AUMENTADA

Fig.1 Graphical user interface for the app cover.

Fig. 2 Graphical user interface for the app thematic content.

Dealing with mathematics is dealing with abstractions, a kind of entity that is conceived in our mind through a cognitive process. Building an Augmented Reality system involves producing a virtual object superimposed on a given space in real time. Some abstractions in mathematics give place to an abstract object, but one that now Augmented Reality could represent. Our aim using AR is to provide our mind with a way to conceive a three-dimensional virtual reality combined with a real-time interaction; this way our perception takes place through the 3D object, a mathematical abstraction taking a visual and tangible form.

The first part of the application proposes a way to conceive the 3D object (surface) from an original 2D object (curve). A parabola, sine and circular shapes will go through a process of repetition in parallel planes because of the effect of a parameter *k* introduced in the algebraic representation of each curve. We are looking for the activation of cognitive processes in the student enabling the development of the spatial visualization skill. Figures 3, 4 and 5 show the considered shapes in the app and their associated algebraic representations.

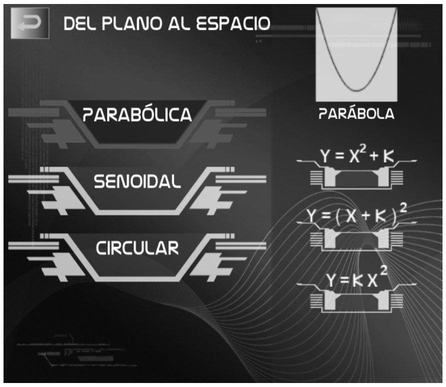


Fig. 3 Parabolic shape and its associated algebraic representations.

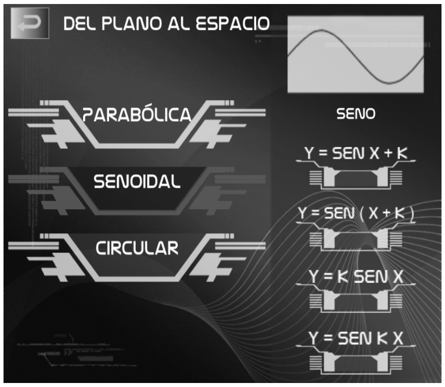


Fig. 4 Sine shape and its associated algebraic representations.

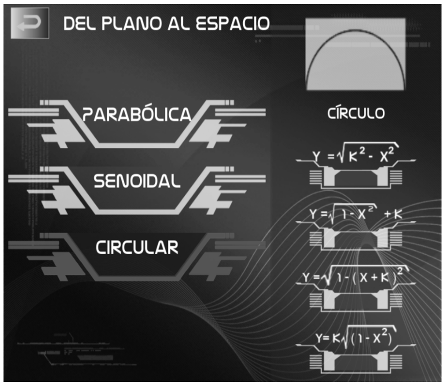


Fig. 5 Circular shape and its associated algebraic representations.

The intention to include RA could certainly be at the service to provide motivational elements for the learning of mathematics, and this also may involve the improvement of student's interest for this science. However, the purpose we mainly seek in our projects is to focus on the didactic benefit of providing a useful knowledge with a new sense of accessibility for a learning experience.

This perspective suggests a new role of technological resources in the learning process. Several authors have been producing appropriate frameworks to study the benefits of didactics involving the use of technology resources. Our perspective shares the conception of digital technology acting as a *mediator* between the user and the mathematical knowledge. The purpose of the interaction between user and technology should favor a co-action that promotes a thinking process that affects learning. Technology should act as a kind of *cognitive partner* for the student, fostering mathematical understanding. Our perspective about digital technology takes into account the knowledge that the discipline of Mathematics Education organizes now in cognitive and sociocultural theories [8, 9 & 10]

3 Introducing the AR Application in in Calculus I

During spring semester 2014 we had the opportunity to work with students in the Calculus I course in charge of one of the authors at Campus Monterrey. We designed an activity to perform in one class, during an hour and a half. The design considered working in teams of 3 students each. All the teams had their interaction with the AR Application from 2D to 3D because we provided them with tablets that were prepared before in order to complete the experience at classroom. The activity was videotaped and each particular team had a recorder functioning to capture the talk they had during the activity. This information has not been yet analyzed but we like to share in this paper the final part of the activity, when the students answered a brief survey we gave them in paper.

The survey we applied is based in the one we report in VARE before, when we interviewed the students that were invited to the use of the Application in May 2013. We wanted to confirm if the information collected then still is given now, with the students in Calculus I 2014 class. We act according with a qualitative case study, certainly now we will no offer the complete an in-depth description and analysis of the case but, the information given on the survey is quite useful. We decided to share it as part of the data collection involving the multiple sources of information that the case study should report [11 & 12]. Figure 6 shows students using the app during the course of Calculus I.

Fig. 6 Students using the app during the course of Calculus I.

Next we will give the review of such information involving a five-question survey that was generated and asked to 20 students. The survey contained open-ended questions that address the interaction between respondents and the application, its usability and its content.

A descriptive analysis of participants' answers has been made and it's presented below accompanied by the corresponding question.

1. Do you consider this application would help you develop any mathematical skill? Which skill would it help you develop?

After asking students if the application could help them develop any mathematical skill, 33% of respondents affirm that it was useful to have a better spatial view of mathematics while 28% think the application could be an alternative way to interact with math. Another 22% are sure that the application helps them to get a real perspective of mathematics, while the last 33% think the application helps the students to better understand and interpret plots and graphics.

2. Could you mention any advantage of the fact that the application is available in an electronic tablet?

Concerning the advantage of having the application available in tablet, 60% of participants think portability and practicality are some advantages, while another 15% think user comfort and the ease of use because of their previous experience with the device are advantageous features. 10% of respondents think the easy access to tablets is favorable, and the same amount of students (10%) supports the idea of using the application as an alternative tool because of its interactivity. 5% think there's no advantage at all.

3. Do you think the use of this application could allow learning? Did you learn something using it?

About the learning perceived, 30% think the use of this application is useful to recognize forms, figures and mathematical structures in nature, while 25% appreciate the 3D visualization that the app generates. 20% of participants think the use of the application helps to differentiate between function types (parabolas, circular and sine), 15% favor the way the app interacts with the content, while the other 10% affirm the use of the application contributes to learn tech development, programming and graphics.

4. Please tell us what you think about the interface and the way to interact with the application.

After asking about the interaction interface, 59% of the students think the application is easy to use and understand, 18% favor the attractive graphical user interface and 14% mention the application need a little tweak, as it's functional but it still has some bugs that can be fixed. 9% of participants express the need to have a shortcut to take a screenshot, while some others (2) would prefer an iOS environment instead of Android.

5. Do you think this application could be useful to acquire knowledge in your career (field of work)? Please give us an example.

Finally, the students were asked if the use of application could be useful in their field of work, and there were many different responses, mostly because all students are in multiple careers. Some think 3D graphics treatment could be achievable with the application, or to establish a link between mathematics and technology. Some other think the application could be useful in acoustics, to observe sound waves, or to better view and understand blueprints. Other students think it could be useful to analyze logistics procedures, or some measurements in industrial machines, to see the behavior of a certain reaction or as an editing software of some sort for their needs, as a visual programming tool or just to use it along the teaching of mathematics. A single student answered the application was not useful at all in their career (Economy & Laws). Figure 7 shows students working during the case study.

Fig. 7 Students working during the case study.

4 Concluding remarks

Mathematics is highlighted because of the abstract nature of their notions and concepts, and the cognitive and learning difficulties that accompany its presentation and understanding confirm a didactic problem that deserve to be working on. Augmented Reality technology stand out for its potential as an attractive alternative when working to create real Mathematics content for students. The abstraction feature of Mathematics feels reachable when a virtual object stands in front of our own space and time reality. Data obtained promise that our work deserves to continue. Improvement of mathematics education should consider the integration of emergent technologies as Augmented Reality.

The current technological development asks for its use in education, and mathematics education has a great opportunity to be fortified. The interdisciplinary team that worked on the first call of Novus 2012 at Tecnológico de Monterrey has been expanding with graduate students, and continues to seek the means to consolidate as TEAM: Educational Technology for Learning Mathematics. Today we continue looking for the ways to overcome our attitude of innovation based in research education working with students of our Educational Innovation Doctorate Program.

We are aware that innovation in education could be considered as such by the mere fact of using technology. But being responsible for an attitude of innovation in the use of digital technology in math education requires a constant updating in new technology development and a continuous attitude of educational researcher. The educational innovation that provides TEAM interested is concerned with curriculum innovation. In this sense, the content of learning and the means by which we learn it are always being subjected to a continuous questioning. Educational research is becoming a way of consolidating this TEAM.

5 Acknowledgements

Tecnológico de Monterrey, Campus Monterrey, has supported this project. Authors give a special acknowledge to NOVUS Call for Innovation for assisted the development of our projects.

6 References

[1] Patricia Salinas, & Juan Antonio Alanís. “Hacia un nuevo paradigma en la enseñanza del Cálculo”; Revista de Latinoamericana de Investigación en Matemática Educativa, 12, 3, 355-382, 2009.

[2] Juan Antonio Alanís & Patricia Salinas. “Cálculo de una variable: Acercamientos newtoniano y leibniziano integrados didácticamente”; El Cálculo y su Enseñanza, 2, 1-14, 2010.

[3] Patricia Salinas, Juan Antonio Alanís & Ricardo Pulido. “Cálculo de una variable: Reconstrucción para el aprendizaje y la enseñanza”; DIDAC, 62-69, 2011.

[4] Patricia Salinas, Juan Antonio Alanís, Ricardo Pulido, Francisco Santos, Julio César Escobedo, & José Luis Garza. “Cálculo aplicado: Competencias matemáticas a través de contextos (Tomo 1)”. Cengage Learning, 2012.

[5] Patricia Salinas, Juan Antonio Alanís, Ricardo Pulido, Francisco Santos, Julio César Escobedo, & José Luis Garza. “Cálculo aplicado: Competencias matemáticas a través de contextos (Tomo 2)”. Cengage Learning, 2012.

[6] Patricia Salinas, Juan Antonio Alanís, Ricardo Pulido, Francisco Santos, Julio César Escobedo, & José Luis Garza. “Cálculo aplicado: Competencias matemáticas a través de contextos (Tomo 3)”. Cengage Learning, 2013.

[7] Patricia Salinas, Eduardo González-Mendívil, Eliud Quintero, Horacio Ríos, Héctor Ramírez & Sergio Morales. “The development of a didactic prototype for the learning of Mathematics through Augmented Reality”; Procedia Computer Science, 25, 62-70, 2013, doi: 10.1016/j.procs.2013.11.008

[8] Raymond Duval. “Un tema crucial en la educación matemática: La habilidad para cambiar el registro representación”; La Gaceta de la Real Sociedad Matemática Española, 9, 1, 143–168, 2006.

[9] Luis Moreno-Armella & Stephen Hegedus. “Co-action with digital technologies”, ZDM, 41, 4, 505–519, 2009.

[10] Luis Moreno-Armella, and Bharath Sriraman. “The articulation of symbol and mediation in mathematics education”; ZDM, 37, 6, 476–486, 2005.

[11] John Creswell. “Qualitative inquiry and research design: Choosing among five approaches (2nd. Edition)”. SAGE, 2007.

[12] Sharan Merriam. “Qualitative research: A guide to design and implementation”. Jossey Bass, 2009.

Global Perspective of Security Breaches in Facebook

Kanwalinderjit Kaur Gagneja
Dept. of Computer Science
Southern Oregon University
Ashland, OR, USA

Abstract— This paper presents, what most people think of when they hear someone's Facebook was hacked? We will present some outdated methods of hacking Facebook that no longer work and how Facebook solved these vulnerabilities. In this paper we also present the current methods of hacking Facebook and how you can protect yourself. When someone says their Facebook had been hacked, what do they usually mean? Nine times out of ten, they left their account logged on around friends who posted a prank status update, or chose a funny picture for the victim's profile picture. This, of course, is not hacking. The other tenth of the time, someone's account has actually been hacked and someone remotely has access to their account.

Keywords—spoofing; keylooger; firesheep; phishing; hackers; evesdroppers;

I. INTRODUCTION

There are a number of social networking sites over the Internet, such as Instagram, InkedIn, Flickr, Facebook, Google+, Twitter, etc. to name a few. Facebook is the largest social networking site in the world. It has over a billion registered users. Actually, we think none of us would know someone who is not already registered. So it concerns all of us.

A researcher from University of Vienna, Stefan Stieger, Ph.D in psychology and some of his fellow researchers took a survey of around 300 Facebook users and almost 300 Facebook quitters to know how these two groups differ in terms of cyber psychology, behavior, and social networking [7].

The responses of the participants were recorded to assess their level of concern over issues such as security and privacy, and users' inclination towards cyber addiction.

The disposition of both groups those who kept on with Facebook and those who had quit the Facebook were looked into. They surveyed them on their behaviors such as outgoingness, neuroticism, amicability, and diligence. From the survey, they found out that about 48.3% of quitters were worried about their security and privacy on Facebook.

The results of above study revealed that the top most reason for the quitter group to quit the Facebook was their concerns over privacy on the social site. So as a user we need to be aware and vigilant about all the possible attacks and how to protect ourselves from them.

The paper is organized as follows. Section II describes about literature review. Section III talks about I was not even logged on! But I have been hacked. Different Facebook hacking techniques are explained in Section IV through section VIII. Finally the paper is concluded.

II. LITERATURE REVIEW

Electronic Frontier Foundation, in 2010 found out that anybody can access information from Facebook profile, although the saved profile is not public [1]. If a user hits 'Like' for some product or service, it creates a "connection" either to product site or on Facebook itself. And such connections or relationships are considered public information on Facebook. Hence, the user's information could be used to be posted on the product or service page hosted on the Facebook [1].

The Facebook is used by Government, federal, state and local agencies to investigate cases to collect evidences to solve some criminal activities. Using Facebook they can find the location, sometimes even could establish the motives, etc…[4]. There are a number of instances when Facebook has readily shared information with government agencies. However, they cannot provide information for private, unopened inboxes those are less than 181 days old, since such accounts need a warrant and a cause by law [6].

May 31, 2010 was observed as Quit Facebook Day. It was an online event. Mainly the users were quitting Facebook because of privacy concerns [5]. It was projected that about 2% of US Facebook users would delete their accounts [10]. Though, only 33,000 users deleted their accounts [3].

Last year it has been identified that Facebook is participating in the PRISM program. Therefore, now Facebook provides information of users from governments all over the world [8].

Data mining over Facebook databases is a real concern. There are number of private companies and individuals unaffiliated with Facebook those are doing data mining over Facebook databases. Lately in 2005, two Massachusetts Institute of Technology (MIT) students downloaded over 70,000 Facebook profiles by running an automated script from four different Universities, that is, Massachusetts Institute of Technology, New York University, University of Oklahoma, and Harvard University. They did it as their research project on Facebook privacy issues [2]. After this incident, Facebook started enhancing security protection for users by building various defense mechanisms to combat phishing and malware [9].

III. I WASN'T EVEN LOGGED ON!?

When someone says their Facebook had been hacked, what do they usually mean? Nine times out of ten, they left their account logged on around friends who posted a prank status update, or chose a funny picture for the victim's profile picture. This, of course, is not hacking. The other tenth of the time, someone's account has actually been hacked and someone remotely has access to their account. Merriam-Webster's dictionary defines a hacker as "a person who secretly gets access to a computer system in order to get information, cause damage, etc…: a person who hacks into a computer system" [11].

Over Facebook's lifetime, it has had multiple vulnerabilities that have been taken advantage of by hackers to either gain people's information, or even just to wreak havoc. These methods take advantage of anything from weaknesses in Facebook's authentication system, to the Facebook user's naivety of internet scams. While some of the methods we cover are outdated and no longer work on Facebook, it is entirely likely that they still work on other websites that Facebook users visit.

For this paper, we researched as many methods for hacking Facebook as we could find and how to implement them. We then tested each of them on our own account, using multiple browsers, a virtual machine, and Wireshark. If they didn't work, we researched what had been done to prevent them from working. If they did work, we researched what methods could be used to stop these attacks. While we could not find how to implement all of the methods used throughout Facebook's existence, we believe we covered almost all of them. The following table I describes different methods with their minimum requirements to implement it.

Table I: Different methods with their requirements

METHOD	REQUIRES
SMS Spoofing	Phone, SMS messaging service
Session Hijacking	Wireshark, Firefox, and a cookie editing add-on like Cookie Manager+.
Phishing	A fake Facebook HTML and PHP file, and a web host
Keylogging	FTP server, Keylogger software
Trusted Friends	Two browsers and four to five fake Facebook profiles

First, we will begin with the outdated methods that no longer work and continue with the current working methods after.

IV. SMS SPOOFING

The first method we tested, which seemed to be the easiest, was SMS Spoofing. SMS spoofing is using a service to send a "spoofed" SMS message, a message that has a falsified sender address. If this method is still possible in the US, it is extremely difficult. We do believe it is possible in other countries. The most common example we saw was India, but most of the information available on the internet lists Australia and New Zealand as the two easiest countries to mask SMS messages.

Fortunately, masking SMS messages is very difficult in the US. Before Facebook solved this issue, it was still very easy to target victims in the US. What the hacker would do is start by going to the victim's profile page, and check if there was a phone number listed in their 'About Me' section. If there was, it meant they had signed up for and enabled Facebook texting. Next they would go to an SMS spoofing site, enter in the victim's phone number to appear as the sender's number and one of the Facebook numbers for a country that SMS spoofing is easier in. Then the hacker could enter anything into the message text and when Facebook received that message, the text would become the victim's status update.

Facebook solved this issue, at least in the US, quite gracefully. They simply asked you to specify which country your cellular service is in and the provider, making it so you couldn't update with messages sent to another country's Facebook number.

V. SESSION HIJACKING

The next method tested that did not work was session hijacking through packet capturing. This method captures packets from network traffic and uses the information in them to hijack someone's session with a website. It requires Wireshark, Firefox, and a cookie editing add-on like Cookie Manager+. First, the hacker would ping www.facebook.com to get its IP address.

Next, they would start a capture on Wireshark and filter it by the IP address you received from the ping. You would then search for a HTTP GET packet with '/index.php'. You would then open the packet and find the cookies. There would have been around 8-10 cookies with varying key, value pairs. The hacker would then open Firefox and open the cookie editor. Next, they would copy the key, value pairs into the cookie editor and add each of the cookies. They would then navigate to Facebook and be logged in as the victim.

Shortly before this problem was fixed, an add-on for Firefox was released called Firesheep. Firesheep would essentially do all of this within Firefox. You would turn it on, it would listen and whenever someone log on to certain sites, it would capture the packets and give you the option to go the site they logged in to and to log in as them. Firesheep was claimed to have been created to bring to light the vulnerabilities in not using HTTPS for cookies. Shortly after Firesheeps creation, many sites began to use encrypted cookies to prevent attacks like these, including Facebook.

VI. PHISHING

The first method tried that was successful was phishing. Phishing is attempting to retrieve information such as usernames, passwords, etc. by pretending to be a trustworthy entity in electronic communication. This method is quite easy to implement. It is accomplished by creating a fake Facebook log in site that collects usernames and passwords. This method requires a fake Facebook HTML and PHP file for phishing, and a webhost for the files. First the hacker would get the files you need and upload them to the webhost site. The hacker would then send the phishing link to the victim(s).

The more victims targeted, the higher the likelihood of success. Once the victim clicks the link, they are sent to a page that looks exactly like the Facebook log in page and is prompted to sign in. If the victim falls for this ruse, the phishing site will record the log in information from them and place it into a file with a name along the lines of 'passes.txt'. The hacker then gets the txt file from the webhost and uses the entire collected log in information, usually spreading the phishing link. This method can be easily defended by being careful which links you follow and always checking the URL of any site you are entering log in information into.

VII. KEYLOGGING

The next method we were able to use was keylogging. A keylogger is either a piece of hardware or software that records keystrokes input into a system. This method can be used with either a hardware keylogger or a software keylogger, and we used a software keylogger in our test since, they are free and used most commonly.

Using a software keylogger also requires a FTP server. They are very easy to use and easy to find. First, the hacker would download a keylogger generator. Next, they would give the generator the process name the keylogger will run as, typically 'server.exe' or something else inconspicuous. They would then designate a FTP server for the keylogger to send log files to and what to name the log files. Next they would designate what size the log files should be before the keylogger uploads them. They would then run the generator which will output a 'keylogger.exe' file, which the hacker then renames to something that the victim is likely to open.

Once the victim runs the keylogger executable, it will run in the background under the name designated by the hacker. It then records all the keystrokes the victim enters until the log file reaches the designated size. The file is then uploaded to the FTP server where it can later be accessed by the hacker at their convenience.

It does all of this without the victim ever seeing the keylogger or knowing it was there. This method is also capable of gathering much more information than just Facebook log in information. It can be used to gain log in information to any site the victim logs into, including banks, to track what they've done searches for and many other purposes. While it's easy to create the keyloggers, they are very easily detectable by the vast majority of anti-viruses, making it unlikely it will even reach many of the victims' computers. In order to even download the keylogger generator, we had to disable my antivirus. Even if the keylogger wasn't detected by an anti-virus, the victim would have to be gullible enough to run an untrustworthy executable file.

Although, we would expect that the people who would go without an anti-virus would also be the people naïve enough to open such a file. This form of attack can be defended against by making sure you have an anti-virus with good reviews with an up to date database and by not opening executable files from sources you don't trust.

VIII. TRUSTED FRIENDS

The last method tested was the 'Trusted Friends' method. This method takes advantage of Facebook's password retrieval system and the victim's willingness to accept friend requests from people they don't know. This method only requires two browsers and four to five fake Facebook profiles. If all goes optimally, only three fake accounts will be needed.

The hacker first makes their fake accounts and attempts to get the victim to accept friend requests from at least three of them. Once the hacker has three accounts on the victim's friend list, they go to the victim's Facebook page and copy the victim's user ID from the URL, (It's the part after 'facebook.com/'). Next, the hacker goes to the Facebook log in page and clicks 'Forgot Password?'.

On the page that loads, the hacker then enters the victim's username and searches it. When it brings up the victim's account with different recovery options, they then click the link 'No longer have access to these?'. Facebook then asks for a new email for verification, which the hacker supplies an email account they have access to. Next Facebook will then either give you the option to 'Recover with the help of trusted friends' or ask for the answer to a security question and will have a link to the "Trusted Friends" option. Once the hacker makes it to this page, they select the three accounts they control as the "trusted friends" and Facebook will send three different verification codes to those accounts. The hacker then uses the second browser to collect the verification codes from the three accounts and enter them into the fields on the first browser. Facebook then sends a verification email to the address the hacker specified and once the hacker activates it, they have full access to the victim's account.

This was probably the most difficult method to complete, especially if we were not targeting our own account. This method would also send far more red flags since the victim's log in information would be changed and they would no longer be able to log in. We have seen fake accounts, which we would assume exist for this purpose, on Facebook very often. A tell-tale sign is an account with an attractive girl for the profile picture, with very few other pictures and friends that are mostly random men from all over. An easy way to protect yourself from an attack like this would be to only accept friend requests from people you know and that would almost completely eliminate the possibility of this kind of attack. If you were paranoid about this kind of attack specifically, you could verify with your friends that the request is them.

If we were actually attempting to use any of these methods to target anyone other than ourselves, we believe the phishing method would be most likely to yield results over the others due to the fact that it would be the easiest to distribute amongst enough people for it to be likely that someone would give their information. The keylogger would be the next easiest since it requires less work, you would simply have to upload it to a file server and make it seem like something people would want to download and just wait for someone without an anti-virus to download and run it.

The "trusted friends" method would most likely be the most difficult to target specific victims. It would however be effective in the scenario we listed earlier. All a hacker would have to do is make multiple fake accounts with attractive women in the profile pictures and spam them to men on the internet. We feel like this would by far have the best results with the "trusted friends" method since men on the internet are more likely to accept a friend request from someone they think is an attractive girl over any other friend request scenario.

As for the outdated methods, we remember when session hijacking was a big problem. It was very common in cafes or any other network that wasn't password protected and had a lot of different users accessing it at any given time. We remember, Firesheep was released in 2010. Everyone was freaking out because they had just found out about Firesheep. The following table II describes the countries affected by the methods and if Facebook solved the problem and if not what we recommend to solve the given method.

Table II

Method	Countries	Resolved by Facebook	Solution by us
SMS Spoofing	India, Australia, New Zealand masking SMS in US difficult	yes	Already solved
Session Hijacking	Anywhere all over world	use encrypted cookies to prevent attacks	use encrypted cookies to prevent attacks
Phishing	worldwide	_	check the URL
Keylogging	worldwide	_	Keep up to date anti-virus
Trusted Friends	worldwide	_	accept friend requests only from people you know

It wasn't long after that most major websites switched to HTTPS for their authentication cookies. We had never heard of the SMS spoofing method though until it was already fixed on Facebook. That being said, this also would have had immense potential for harassing people and, overall, it's good that it has been solved.

IX. Conclusion

In conclusion, Facebook has done their best to protect its users from hackers and people who would gain access to their user's information against the user's will. They have done a very good job with it so far, as all the methods we could find that still worked took advantage of the user and not weaknesses in Facebook itself. While Facebook is doing its best to keep you safe, there are many people out there who are creative and talented at tricking people into giving out their information. Even though our Facebook accounts can almost never be completely safe from a hacker, practicing sensible browsing and common sense on the internet will go a very long way to keep hackers out.

X. Disclaimer

The information in this presentation is for instructional purposes only and is intended to inform others of how easily their Facebook account can be hacked and how they can protect themselves. We tested all of these on our own account, on our own computer using multiple browsers and a virtual machine. Please don't try this at home.

References

1. Esguerra Richard, "A Handy Facebook-to-English Translator | Electronic Frontier Foundation", April, 2010.

2. Harvey Jones, and Hiram Soltren José, "Facebook: Threats to Privacy", Cambridge, MA, Ethics and Law on the Electronic Frontier, Fall 2005.

3. Jemima Kiss, "Facebook: Did anyone really quit?", Guardian, London, June 2010.

4. Lynch John, Ellickson Jenny, "Computer Crime and Intellectual Property Section, Obtaining and Using Evidence from Social Networking Sites: Facebook, MySpace, LinkedIn, and more", U.S. Dept. of Justice, 2013.

5. Paul Ian, "It's Quit Facebook Day, Are You Leaving?", PC World, May 2010.

6. Semitsu Junichi P., “From Facebook to Mug Shot: How the Dearth of Social Networking Privacy Rights Revolutionized Online Government Surveillance”, 31 Pace L. Rev. 291, 2011.

7. Stieger Stefan, “Quitting Facebook—What's Behind the New Trend to Leave Social Networks?”, Cyberpsychology, Behavior, and Social Networking Journal, September 2013.

8. Stretch Colin, “Rapport over verzoeken tot gegevensverstrekking van internationale overheden Facebook”, August 2013.

9. Wolens Fred, “Facebook Security Response”, TheIndyChannel, Nov. 2010.

10. Woollacott Emma, “Quit Facebook Day set to be a flop”, TG Daily, May 2010.

11. http://www.merriam-webster.com/dictionary/hacker

A Graduate Merit Scholarship Program at Western Illinois University

John Drea[1] and Martin Maskarinec[2]
[1]Associate Dean, College of Business and Technology, [2]School of Computer Sciences
Western Illinois University

Abstract - *A proposal is made to develop a "graduate merit scholarship" for graduate programs that have available capacity by creating and implementing a scholarship program restricted to high performing students from universities that do not supply graduate students to WIU (both US and international universities.) By adding students that would not otherwise attend WIU to fill available spaces in targeted graduate programs, the proposal estimates that between $11K-$14K in additional marginal revenue could be generated for each additional enrolled student. Implementing this initiative for only seven students within one graduate program would likely increase the performance potential of the graduate program (higher incoming GPAs and standardized test scores) and produce over $100,000 of additional revenue for the income fund.*

1. Background

Like many state universities, Western Illinois University (WIU) faces many budgetary challenges. WIU is faced with declining support from the state and declining enrollments, both pushing revenues downward. On the other side, expenses continue to rise at least at the rate of inflation. (see [1] for an historical view of expenditures and [2] for an historical view of enrollment numbers). These factors put pressure on the budgeting process.

One method of alleviating some of this pressure is to recruit more International students to fill open seats, particularly in our Graduate Programs. However, a major problem exists: (not surprisingly) the tuition is quite high [3], placing it out of reach of some students, and making us less competitive in the eyes of others. Specifically, graduate tuition at WIU is set at a rate higher than undergraduate tuition (even for in-state students) and the out-of-state rate is double that.

Our solution to this was to develop a Graduate Merit Scholarship (GMS) to provide some cost savings to highly qualified International Graduate Students. Obviously, constructing a scholarship program is not new; constructing one that can be approved by the university's administration is another matter. This paper describes the program and the process used to convince an administration concerned with revenue generation that this scholarship program will add revenue, not subtract revenue.

2. GMS Overview

The scholarship program has several key components:

- The scholarship program provides a 25% reduction in the cost of tuition for its recipients.
- The Departments participating in the GMS program must identify the number of empty seats they have available to fill without needing additional resources.
- The recipients must meet stringent academic requirements (GPA as well as standardized test scores).
- The recipients must matriculate from "partner" institutions.
- The recipients must matriculate from a university from which the department has not received students in the past three years.

Let us know look at each one of these in more detail:

2.1 GMS Covers 25% of Tuition Cost

Obviously, a scholarship needs to be a monetary award. A percentage of tuition was chosen because the award may then be adjusted upward without needing to go through an approval process each year. The 25% amount was chosen as a midpoint between instate tuition and out-of-state tuition, which is twice as much as instate. Thus, an argument can be made that the scholarship recipients are still paying more than instate students (who are viewed as getting some of their costs paid for by state support).

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

2.2 Participating Departments Identify the Number of Available Seats

Departments may elect to participate in the GMS program or not. If they do, they must do so in a way that would guarantee that they would not need additional resources to support the students. This is to ensure the additional tuition revenue would not be offset by additional costs (and thus garner less support from the administration).

2.3 The Recipients Must Meet Stringent Academic Requirements

Graduate Merit Scholarship applicants from partner universities would apply for admission into the graduate program and submit a graduate admission examination score (such as the GRE or GMAT). The graduate admission examination score is important to verify the expected performance level of students from undergraduate institutions that we have had little/no previous experience (and without a basis to assess how these students would perform in a graduate degree program at WIU.) Using a graduate admissions examination score for awarding graduate merit scholarships is consistent with how ACT scores are used for undergraduate scholarship programs available at WIU.

It is important to note that not all graduate programs at WIU require a graduate admission exam as part of the admissions process. It is not the intent of this program to change the admission requirements of these programs; rather, the submission of a graduate admission exam score would only be required of those students that also would like to be considered for the Graduate Merit Scholarship.

2.4 The Recipients Must Matriculate From "Partner" Institutions

Both of these requirements were put into place to help ensure participating departments that we were interested in attracting highly qualified students, and not just bringing in additional revenue at the cost of quality.

Each department is required to set its own academic requirements as well. To date, all participating departments have established a GPA threshold. Also, all have established some standardized test score threshold as well. Currently, our departments are using GRE and GMAT scores here.

2.5 The recipients must matriculate from a university from which the department has not received students in the past three years

One of the objections from the administration was that we would just be giving a tuition reduction to students who would have come here anyway (and paid full tuition). This requirement alleviated this a bit by guaranteeing the students were coming from new feeder schools and not from places we have successfully recruited in the past.

It is also important to point out that this was done at the department/program level, not at the university level. Thus, if the Computer Science program has a good relationship with a feeder school, this school's graduates were not eligible for a GMS *in Computer Science*. However, student matriculating from this institution into WIU's MBA program (for example) would be eligible for a GMS.

3. Operationalizing the GMS

The GMS should be as invisible to the student as possible. Thus, the student will submit her/his application in the normal way (with the only addition a graduate examination score, if not required). To the student, this should be the end of the process until s/he is sent a notification of award.

From the university's prospective, the graduate coordinator for each participating program will be responsible for collecting the applications of all applicants qualifying for a GMS. At a prescribed time (usually April 1 for Fall admits and October 1 for Spring admits), the graduate coordinator will assemble all applications and present them to the program's Scholarship Committee. This committee will then be tasked with deciding on which applicants should be awarded a Graduate Merit Scholarship. Of course, if the number of qualifying applicants is less than the number of scholarships which may be awarded, all qualifying applicant will be awarded a GMS and no further action on the part of the Scholarship Committee will be required.

The list of recipients will then be forwarded on to the Budget Office. This office will then have the responsibility of posting the scholarship to the student's account when s/he arrives and registers for classes.

4. Estimating Revenues

The projected costs associated with the program are expected to be minimal. It is critical to recognize that the tuition revenues from the program are all marginal tuition revenues; that is, they are revenues that would not have been received without the presence of the scholarship program. Thus, it is important not to view the program in terms of the $6,000 "expense" of a graduate merit scholarship awarded to a high performing graduate student; rather, the financial viability of the program should be based upon the $12,556.56 in marginal tuition revenue generated by that same student.

5. Estimating Costs

There are two categories of costs associated with the proposed scholarship program: scholarship costs and promotional costs.

The scholarship costs would be supported from the university's income fund, since scholarships would only be issued to students that would not be in a graduate program without the scholarship program. Thus, these costs are completely offset by the increased revenues produced by attracting additional graduate students.

The promotional costs would be expenses related to creating and developing relationships with targeted universities. One of the resources available to WIU is the use of current faculty who often travel back to their home regions over breaks. It is anticipated that many would be willing to call on a university in their home region, make introductions, and distribute literature on WIU graduate programs and the proposed scholarship program for a nominal supplemental contract. Such contacts could also be made by faculty/administrators who participate in international travel for conferences and meetings if the travel is in proximity to a potential international partner.

If six faculty members were paid a $250 stipend (each) to call on a potential partner university, build relationships and distribute information, promotional costs for this activity would be $1500 plus $500 printing ($2000 total). The marginal revenue from one $5K graduate merit scholar in a 33 semester hour graduate program is $14,438.32, so the break-even point for the proposed program is

$$\begin{aligned} BE_{\text{Units}} &= \text{Fixed Costs}/(\text{Tuition} - \text{Variable Costs}) \\ &= \$2000/(\$14{,}438.32 - 0) = 0.14 \end{aligned}$$

Thus, if only one student matriculates to WIU who would not otherwise have come, the GMS program will generate positive revenue.

6. Diversity Benefits

In addition to the financial benefits of the proposal, the university could diversify the international student body by developing partnerships with universities in geographic areas that are currently underrepresented on the WIU campus. As of Fall 2011, the university enrolled 353 international students on the Macomb campus (both graduate and undergraduate) (source: 2011 WIU Fact Book, p. 52). Of these, over half come from two countries: India and Saudi Arabia. It is noteworthy, however, that students from these two countries are not evenly distributed across all programs, i.e., Computer Science has a large number of Indian students in their graduate program, while the MBA program has very few Indian students.

Most graduate programs make considerable use of classroom discussion and learning through collaborative projects. This makes graduate education an important beneficiary from a broader representation of countries among the graduate student population. For the MBA program, additional students from India, Brazil, Mexico, South Korea, sub-Saharan Africa, Eastern Europe, and the Caribbean would enrich the classroom experiences of all of our students by providing different perspectives on the world at large.

7. Conclusion

A Graduate Merit Scholarship program has been developed that allows for strong students matriculating from partner institutions to receive a reduction in tuition. To do this, the authors have developed a plan that would apply only to new students (thus avoiding the revenue reduction argument if it were to apply to students we would receive anyway). Further work must be done to build the relationships with and to partner institutions to ensure a flow of gifted and high-achieving students into our graduate programs.

References

[1] "Operating Expenditures", Appropriated Funds Budget Fiscal Year 2014, Western Illinois University, Page 8.

[2] "Headcount Enrollment by Class Level and Location, Fall 2009-2013", The 2013 Western Illinois University Fact Book, page 31.

[3] "Graduate Students - Plan P for FY15", Tuition and Fees and Other Costs, Western Illinois University, Available at http://www.wiu.edu/vpas/business_services/tuition/graduates/planP.php.

Learner Centered Schemes in the Digital Age

M. Estabrooks
Applied Science Department, Red Deer College
Red Deer, Alberta, Canada

Abstract - *For many educators today, the internet represents an extremely useful means for the delivery of educational content in a timely manner with minimal cost. Indeed, institutional officials feel that the entire curriculum and assessment may also be carried out via the internet with minimal or no human intervention.*

As our experiences with digital technology evolve, some observers feel that there is a down side to extensive use of these digital devices. The author feels that digital technology often exploits various addictive tendencies of the brain and attention is diverted away from the desired task such as skill development in mathematics.

In this article, we present the results of a study involving a lecture based delivery model and another model that focuses on classroom engagement with face to face interaction and intervention techniques. This study suggests that the instructor role in the new digital age becomes more that of a coach much as in sports.

1 Introduction

Digital technology has proved to be a great boon to individual expression and social awareness as well as providing a new era for accessing information unprecedented in history. Perhaps no sector of our socioeconomic society has benefited more than the education component. With its interactive capability and the delivery of educational content in a 24-7 format, it promises that education and learning can now be designed to fit into the learner's time and space constraints and thus ensure greater success in achieving one's educational goals.

Despite this, the author is of the opinion that extensive use of digital technology, and especially social media tends to promote a withdrawal of face to face engagement brought about in part by the uncertainties of f2f interactive communication. Furthermore, this leads to disengagement of the user, preferring to interact in a minimal manner and does affect the degree to which the learner can become engaged in the subject matter in a classroom situation.

This paper will focus on College level statistics and efforts by the author to address the very diverse needs of students in several sections of the course. Success in statistics requires a medium level of algebra and math skills accompanied by an appropriate level of reading and study skills. The lack of an appropriate level of skills in these areas, coupled with constraining factors such as extra-curricular activities, outside work and addiction to social media all provide a complex interplay of variables which play significant role in relation to success in the educational setting. Without appropriate schemes in place to address the issue of student success with these constraining factors in place, the instructor is faced with high dropout and failure rates in the class

This then begs the question as to how an instructor or professor is able to address the diverse needs of such students and at the same time ensuring greater student success without sacrificing course content and grade inflation.

2 Research Rationale

During the period of 1980 thru 1992 the class averages in statistics were somewhat constant averaging 66-67 throughout the period. However, beginning in the late 1990's that class averages began falling with each year (Figure 1). The trend appeared to have no explanation from a statistical perspective as many factors were at play and the phenomena appeared to have no direct link other than lower entry level skills.

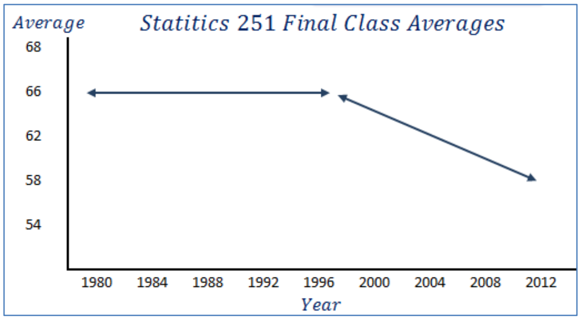


Figure 1

Faced with high dropout rates and high failure rates, a strategy was formulated to address these issues. The effort began by identifying those aspects of the teaching-learning process that are within the domain of the instructor/professor.

First consideration was the nature of the text book. The author feels that traditional mathematics and statistics textbooks often have intimidating page layouts filled with formulas and very tight text layouts. Moreover the coverage is often so vast as to be very daunting to the beginner level student.

To address many of these limitations, the author developed a 320 page "class workbook" intended to provide a focus of work activity for the student. This workbook features minimal theory with numerous word problems. The workbook layout provided space for students to demonstrate specific procedures utilizing paper and pencil techniques deemed necessary for proper problem solution.

Copyright © 2014 CSREA Press, ISBN: 1-60132-271-2; Printed in the United States of America

Secondly, negotiations with a publisher resulted in the adoption of a customized text book with internet access at a substantially less cost than previously was available. The on-line content provided a means for self-testing with appropriate feedback allowing the student to undertake a more or less independent approach to learning and allowing self-assessment to gauge one's skill level.

Thirdly, the author developed thirty nine on line videos of 10 to 12 minutes duration, corresponding to 33 class hours. Each video addressed a topic that was covered in a class and these same questions appeared in the workbook thus enabling the student to do the workbook problems at home using these on line videos. The videos were made available in a 24-7 time frame in order that students are able to access the video content at any time.

Finally, on a weekly basis, the author provided a detailed study guide specifying which videos to watch, workbook problems to be completed and timelines for on-line assignments available. The tenor of the comments was such as to provide information and encouragement to the learner.

3 Methodology

During 2010, two sections of beginner level statistics were identified to become part of a study to assess two classroom management schemes.

Group A consisted of one section conducted utilizing a traditional approach with primary lecture techniques involving blackboard and PowerPoint presentations. The classroom workbook was used for a 10-15 minute problem solving session in each class..

Group B consisted of a second section of statistics conducted with the extensive use of videos, accompanied by very detailed weekly summaries of what sections of the textbook to read, problems to do and videos to watch etc. The class was conducted with a 15-20 minute presentation of key content material and problem solving techniques followed by 35 minutes of problem solving in groups with face to face interaction. During this time, the instructor was monitoring progress and interacting with students in more or less a coaching role, attempting to connect with each students and providing encouragement and help.

At the end of the semester in 2010 the final scores for each section were complied, and these results are displayed in the histogram in the figure labelled Figure 2 here.

A glance at the histogram illustrated at the top of Figure 2 suggests an extremely varied distribution with a mean of 49. This distribution may be considered typical of a presentation feature involving primarily a lecture format.

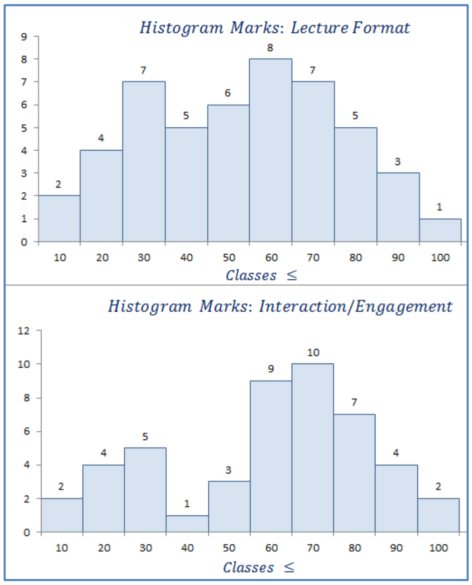


Figure 2

The bottom half of the Figure 2 depicts the histogram if marks for Group B involving classroom engagement, face to face interaction with instructor actively involved in the classroom. The most distinguishing feature of this histogram is the bimodal nature of the data suggesting two sub-populations present. Whereas the average for this class was 56-58%, the appearance of the two cluster points of 27 and 67 suggests that various factors were operating within the "populations" involved

4 Discussion

It is considered difficult to apply standard statistical methods to the data in question because the same instructor was involved in both deliver schemes and as such, it was difficult to block out instructor behaviors. Furthermore, one cannot treat the sections as being random samples from a population as the two sections were simply the only sections held during the semester.

In an attempt to identify potential variables at play in this study, questionnaires were designed and administered by phone or with personal interview follow-up. On the basis of the feedback obtained, the following "factors" were identified as significant:

- Lack of an appropriate math & study skills ;
- Number of courses enrolled;
- Outside work, often 16-24 hrs per week;
- Time spent on student activities & sports;
- Social media, internet & cell phone use;
- Personal issues, family, relationships, money etc;
- Timing of the class, mid-day vs late day times;

It seems obvious that Group B benefited significantly from the management scheme involving classroom engagement strategies and face to face interaction etc.. Most notable in this regard is the degree to which students began to help each other in problem solving situations as well as making suggestions regarding which videos to view, which learning resources to use for study etc. Overall, there appeared to be a greater sense of satisfaction with what was happening in the classroom. Indeed on several occasions students would comment to the entire class that s/he was enjoying the classroom experiences.

5 Observations

This study revealed the rapidly changing world of digital technology and its impact in an educational setting. Considerable attention must be given to designing learning situations to meet the changing needs of our students. The following observations and speculations may be considered:

- Students less willing to assume responsibility for his/her learning;
- Greater feeling of entitlement & expectations;
- Greater connectedness to cell phones and social media to the extent that these devices with their rapid response and continuous engagement features tend to capture students attention subverting attempts to focus on one line of thought for a short period of time;
- Difficulty with comprehension and understanding word problems, especially those involving more than 30 to 40 words;
- Apparent loss of the art of human communication and inability to carry on a conversation in an interactive manner;
- Apparent inability to focus on the subject matter at hand and put thoughts and ideas together in a cohesive manner.
- Apparent loss of writing and critical thinking skills ;

Currently, efforts are underway to address these issues by working more closely with students on a one to one basis in order to effect greater commitment to success in the course.

6 Conclusions

Based on the results of this study, the author is of the opinion that the traditional lecture format utilized by post-secondary institutions does not benefit the majority of students today. The vast majority of students at the post-secondary level come to us with such an extremely varied background so as to suggest that there is not one solution that fits all situations.

Rather it appears that any solution to addressing the needs of today's learner should involve a diversity of schemes a few of which are outlined above. This may take many forms including appropriately designed digital media as well as group work, problem solving sessions, as well as project based learning with significant instructional leadership. Such a setting allows the instructor to better interact with the students and provide greater face to face interaction while monitoring individual student progress.

The author suggests that a new paradigm for addressing student learning needs in the digital age is evolving. The model places the instructor role to involve monitoring student progress, motivating and guiding the learner in a process involving face to face interaction and intervention techniques. As such the instructor's role becomes more that of a coach, much as in sports.

7 References

[1] Carr, Nicholas, *The Shallows: What the internet is Doing to Our Brains*, Norton & Company, 2011

Using Undergraduates as Teaching Assistants at Harbin Institute of Technology

Dalie Sun[1], Hongli Zhang[2], Kuanquan Wang[3]
School of Computer, Harbin Institute of Technology, Harbin, China, 150001
1 sdl@hit.edu.cn, 2 zhanghongli@hit.edu.cn, 3 wangkq@hit.edu.cn
Contact author : Dalie Sun

Abstract: For many years the programming courses in Harbin Institute of Technology are so inefficient that many students can’t write programs efficiently, ever though they have got high grades. One of the reasons is that we are lack of instructional supporting, so that our students can’t get necessary help timely. Using undergraduates, who are good at programming, as teaching assistants not only deal with the problem, but also help the undergraduates themselves doing better. And it brings an efficient way for teaching of the courses.

Keyword: undergraduates, teaching assistants, course, programming, students

1. Introduction

This program began in 2008 when we developed ACM ICPC(ACM International Collegiate Programming Contest) in Harbin Institute of Technology. In 2008 we were designated to hold a post of coach of HIT ACM ICPC. The first task we faced was how to get higher prize in regional and how to develop the contest all over our institute. What we have done firstly was to introduce the contest into the relative courses, such as C programming language. We integrated the contest problems and concepts into the course as the part of itself, and guided the fresh students to do contest problems as exercise and homework.

But it was not enough. We should get touch of these fresh students more often and more close. How to do? By analyzing all process of the course, we found the breach on teaching assistant (TA). For an example, in the course of 13SD03100300 which is opened in fall for fresh students, all teaching assistants were undertaken by graduates. As teaching assistants, they should contact often with the fresh students to give their support face-to-face and step by step every week or every day. But their performance was not so good. With the license of our dean, We tried to replace half of teaching assistants in course 13SD03100300 (High Language Programming) with our undergraduate students who had more than two years of experience on ACM ICPC. It was so successful that in next year all teaching assistants were undertook by these undergraduates. And then we expanded this program to other courses, such as 13SD03000510(C Language Programming), 08N1031010 (Formal Languages and Automata), 08T1031050(Operating System), etc.

This program not only made the ACM ICPC known well among the students but also help us to find a way for teaching and learning. Maybe it is more important for us to get our courses more efficient.

2. Related Courses

(1) List of Courses

We apply the program in following courses. These courses are almost opened at the first or second semester for first year students.

- 13SD03100300, High Language Programming, is set up for computer major students at their first semester. It is an obligatory course for Computer major students. There are about 260 new students in our School of Computer Science and Technology every year.
- 13SD37000200, High Language Programing, is set up for software engineering major students at their first semester. It is an obligatory course for software engineering major students. There are about 160 new students in our Software Engineering Department every year.
- 13SD03000510, C Language Programming, is set up for all major students at their second semester, except computer major, software engineering major, and architecture major. It is an obligatory course for all students for which it is opened. Every year there are about 3000 students who register this course.

• C000300500, Primary Programming Practice, is set up for all students who have programming experience ever. There are about 200 students to register this course every year.

(2) Instructional Support of Courses.

In a long term, it is a big crux that there is almost no any instructional support in courses, especially in these programming courses. In 2005, Graduate School in HIT implemented a program that every scholarship graduates must experience teaching assistant work at least once during their first year. At the same time, they made a restriction on the number of teaching assistants in each course. The situation of the instructional support of courses is shown as table 1.

Table 1 Number of TAs in Courses in 2007

Course	Major	Number of students	Number of TAs
13SD03100300 High Language Programming	Computer Science and Technology	240	16
13SD37000200 High Language Programming	Software Engineering	180	4
13SD03000510 C Language Programming	General(not CS, SE)	3000	40

Before 2007 the number of teaching assistants in course 13SD03100300 is also 4. In order to strengthen the support of programming course, CS department deploy more graduates that 1 graduate correspond 15 students. Only this course can break through the restriction with the license of Graduate School.

3. Selecting Teaching Assistants

Up to now, almost all teaching assistants are chosen from the contestants of ICPC. There are many kind of programming contests for our undergraduates to take part in which is titled with ACM ICPC. From the highest level to the lowest, the list of these contests is shown as bellow.

- Annual ACM International Collegiate Programming Contest World Finals. It is held in spring once a year. Almost all teams(contestants) are chosen from every regional who is among the best ones at programming in their school, their university, their country and their continent.
- Regionals. It is held in the autumn once a year before the World Finals. It is partitioned into 6 regions, such as Africa and the Middle East, Asia, Europe, Latin America, North America and South Pacific. There are several sites or more in each region. For example, there are 15 sites in Asia(5 sites in China).
- Provincial Programming Contest. There are about 20 provincial programming contests held in the spring in China every year. In each contest there are about 160 teams(3 contestants/team) competing with each other at a hall.
- University Programming Contest. In China there are many university programming contests held once or twice every year. We hold our university programming contest twice a year. About 500 students including few graduates take part in the contests each year. Most of them are first-year undergraduates. Some of them are second-year undergraduates.

There are about 50 contestants chosen from our university contest to take part in higher level contests, such as Heilongjiang Provincial Programming Contest, Northeast(China) Programming Contest and Asia Regionals. They got at least Bronze Medal on these contests.

For course 13SD03100300 and 13SD37000200, we assign one teaching assistant for about 15 students. So we need 18(260/15) and 11(160/15) TAs for these two courses. It is enough for us to select the suitable candidates from the above good contestants.

For course 13SD03000510, we need about 200 teaching assistants meanwhile. Surely we can choose these teaching assistants from the contestants of our university programming contest. But we don’t expand the program all over the course. We have started the program in small scale including about 600 students. It is just an experiment for us to get supporting data by which we can convince our administrators.

4. Training Teaching Assistants

As a teaching assistant, he should meet the following qualities.

- Be good at programming. That means he knows algorithms well so that he can quickly find out the best algorithm to deal with the problem. And he is good at coding so that he can write the programs fast and efficiently.
- Be responsible. Though we pay reward for his work, it is very important for him to do as better as he can.
- Be familiar with teaching skills. Maybe they are very skilled at programming, but the fresh students are not. What they should do is make these fresh students do as better as they do. It is necessary for them to learn about some teaching skills, such as how to express, how to grade, etc.

There is a great difference between our program and what the conferences describe. Many of our students are weak at programming even though they got high grade of the course. So we must pay more attention to the programming capability of our teaching assistants at first. We train them mainly by holding programming contests and seminars described as bellow.

- Weekly programming contest. Students have improved their programming skills when they take part in the ACM ICPC. We hold online contest once a week almost all over semester to train students step by step. It is operated by 3-year or 4-year undergraduates who have at least two year experience of ACM ICPC. In another words they are also teaching assistants. They run the contests all by themselves, such as determining problems and Q&A online.
- Weekly seminar. The seminar is held after each online contest and hosted by our higher contestants also, working as teaching assistants. They play the role of lectures, giving short explain about the problems, responding the question of the students and discussing with them.
- Summer training. We hold summer training every year from the first day to last day of summer holiday. Yes, all students who taking part in summer training can’t enjoy their summer holiday. There are about 70 students at beginning, and about 50 students remaining at end. We design the training schedule and use the ‘old contestants’ as teaching assistants who have outstanding performance in programming. These teaching assistants take over 80 percent of training work.

The whole summer training continues for 5 or 6 weeks all over the summer holiday. We work 6 days each week from Monday to Saturday. Every morning we give a lecture of algorithms introduction. After noon we do exercises or hold a contest. In evening we review and discuss the algorithms and problems. Lecturer takes the work of algorithm introduction on morning. And teaching assistants play the role of exercises/contests and discussion. They also take some work of morning lecture. The first week schedule is shown in table 2.

Table 2 First Week Schedule

	Morning	Afternoon	Evening
Monday	Data structure Stack/queue/tree	Exercises: Dichotomy	Review
Tuesday	Greedy algorithm Maximum/minimum heap	Contest 12:00-17:00 Greedy algorithm	Discussion 18:30-21:00 Contest problems
Wednesday	Introduction : Standard template library	Contest 12:00-17:00 Complex	Discussion 18:30-21:00 Contest problems
Thursday	Dynamic programming Basic	Exercise : Problems/acm.hit.edu.cn	Review & discussion
Friday	DP: Monotonous queue	Contest 12:00-17:00 DP	Discussion: Problem solving report
Saturday	DP: Memory search	Exercise : Test & problems	Discussion : Conclusion of DP
Sunday	Rest	Rest	Rest

5. Using Teaching Assistants

In this paper we describe the work of our teaching assistants program by taking the course 13SD03100300 as an example. The situation on course 13SD37000200 and 13SD03000510 is very similar besides (6).

(1) *Structure*. All students are divided into two classes which belong to two lecturers. All students in same class attend the same lecture of the same lecturer. Each class is divided into two groups of which all students are instructed by the same group of TA. The specific description is shown as table 3.

Table 3 Teaching Structure of Course

13SD03100300 High Language Programming Students : 280	Class 1 Students : 137 Lecturer : Wang	Group 1 Students : 60 Leader : A	Number of TA : 4 Instruct/Grade/Q&A
		Group 2 Students : 77 Leader : B	Number of TA : 5 Instruct/Grade/Q&A
	Class 2 Students : 143 Lecturer : Zhang	Group 3 Students : 87 Leader : C	Number of TA : 6 Instruct/Grade/Q&A
		Group 4 Students : 56 Leader : D	Number of TA : 4 Instruct/Grade/Q&A

(2) *Instruction*. The total teaching hour of course is 48 hours, including 30 lecture hours and 18 coding hours. Teaching assistants don’t need attend the lecture. The lecturer is in charge of all lectures. Teaching assistants are responsible for all coding instruction. They spend three hours every week on instructing students to write and debug programs. At this point all TAs in the same group work together for the same group of students. They go around in the class room, to answer the questions, to inquire about if someone gets in trouble.

At beginning they meet many problems and heavy work. The fresh students even don’t know how to play keyboard. Someone plays keyboard with only two fingers - two forefingers. Teaching assistants must correct these bad habits at first and set a good example for them to imitate. Most of students don’t know how to debug program – no idea how to find errors and correct them. Teaching assistants guide them debug their programs step by step.

(3) *Homework & Project.* There are five homework and two projects waiting for teaching assistants to review and check. Each teaching assistant is responsible for about fifteen students. They review the homework, checking whether correct and whether suspecting plagiarism, finally grading. At first two times they write more detailed reviews, later they just give a rough comment. It takes them about 6 hours at average on this work.

It takes them much time to review the projects. They check the projects by discussing with every student. With the submitted materials they propose questions for student to answer. Sometimes they delete several lines from the program to check whether the student really write the program by himself. During this time they spend two hours at average to talk about with one student every day.

(4) *Online Q&A.* Teaching assistants are responsible for online Q&A. Each group assign to a person on duty every day. Maybe someone likes to stroll online and do the work on the way. This is a relatively easy work.

(5) *Grading*. It is a very challenging work for our teaching assistants. Though the criteria are very clear, there is much difficulty on implementation. We can’t get the exact value of a program by putting it on a weight scale. So they put all students’ work together for comparison with each other and finally give a relative grading. They write a grading report meanwhile for replying complaints of students later.

(6) *Mentoring*. We assign an additional task to all teaching assistants – to be mentor of the freshmen. Each teaching assistant is responsible for about 15 students.

According to the schedule, he should know of every one of his 15 students at first week, get separate conversation with each one before third week, inquiry and write abbreviated record of them every week. After the end of the course every teaching assistant submit a summary report including the record of his students.

(7) *Lecturing.* We make a further attempt in the course C000300500. The course hour is 30 and all is coding hour. That is, we bring our students talking about algorithms and writing programs in classroom with one computer per student. Firstly, we propose a real problem. Secondly, we talk about the algorithm suitable for the problem. Finally, we write the program to deal with the problem. Most of the problems are chosen from ACM ICPC online judge. Our contestants are very familiar with these kind of work. So we let them play the role of lecturer in this course.

4. Advantage

Just like the program mentioned in the conferences, using undergraduates as teaching assistants has many advantages. We agree with the description of the conferences. Because of different situation, it has special meanings in Harbin Institute of Technology.

- Undergraduates are more efficient than graduates. These undergraduates are chosen from our ACM ICPC contestants who have experienced at least Provincial Programming Contest. They are good at algorithms and programming. Most of them are third year or two year undergraduates. What happened in their fresh time is still vivid in their memory. They are easier to deal with the the problems the fresh students faced. They are so familiar with the course that they can find out problem at a glance without waiting for the students asking them.

- Undergraduates are more responsible for the work. For most of them this is their first job in their lives, even though with a little salary. It is interesting enough to make them excited. We have really felt it from their wonderful performance.

- Using undergraduates as teaching assistants establish a community of undergraduates in different grade and different school. It is very helpful for them. In our university, all one year undergraduates are living another campus apart. At their first year they have seldom chance to touch and exchange with the seniors. In the program meet their tutors every week and often discuss with them. It is going to continue after the course end and loop next year.

- Undergraduates have improved themselves greatly by this program. They not only developed programming skills, but also developed many skills such as expressing skills, exchanging skills, understanding, communicating and so on. Especially, they have got considerable progress in lecturing. One of our students suffering from stuttering disorder, after playing the role of teaching assistant two times in the course C000300500, he can give a very fluent presentation now.

- Using undergraduates as teaching assistants have improved programming of our students greatly. Before we start the program, many lecturers who take follow-up courses complained that they can’t exchange with students because of their poor programming. Now they say that they can talk about the courses deeply with students.

5. Conclusion

When we start to use undergraduates as teaching assistants, we didn’t understand that we can get so great benefit. Through this program, we not only have developed ACM ICPC in our university, but also have made our programming courses more efficient. The most important is that we get a new style of teaching.

What we want to do in future is to develop the program in all programming courses, especially in 13SD03000510, C Language Programming. It is a big challenge for us to choose 200 graduates as teaching assistants.

References

1. Eric Roberts, John Lilly, and Bryan Rollins.Using Undergraduates as Teaching Assistants in Introductory Programming Courses:An Update on the Stanford Experience

2. Stuart Reges. Using Undergraduates as Teaching Assistants at a State University

3. Dalie Sun, Lixin Wang, Zhaonian Zou. ACM ICPC in China-Learning From Contests.

AGILE: HIGHER EDUCATING STEM COLLEGE STUDENTS

Stacey Franklin Jones, D.Sc.
Elizabeth City State University

Abstract

The general consensus is that the demand for science, technology, engineering and mathematics (STEM) workers has and will continue to increase in the foreseeable future. However, the misalignment between what employers need and what skills are taught and delivered has been expressed as a critical problem for U.S. competitiveness.[i] *Exposure during those formative college years to a broader application of the tenets of the Agile methodology which is gaining significant traction in industry and government, holds the promise of better preparation of those entering the workforce. This approach may be a pivotal element of a solution to higher educate STEM students.*

Majoring in STEM – Necessary, but not Sufficient

While the statistics vary, the general consensus is that the demand for science, technology, engineering and mathematics (STEM) workers has and will continue to increase in the foreseeable future. This is good news for students who have endured the rigor of college level study to earn STEM degrees. However, the lack of alignment between what employers need and what skills are taught and delivered has become a critical problem for U.S. competitiveness.[ii] Project based learning has emerged as a promising educational approach to mastering skills for the workforce. In fact, there are universities that have completely abandoned their traditional lecture-based pedagogy.[iii] Also topping a list of the Five Ways to Better Prepare Students for Careers is "teamwork". These components – project based learning and teamwork - considered together yield statements such as "Employers want people who understand how to manage a project, how to deliver a product on time and on budget, and how to work in teams with little or no oversight".[iv]

Agility

Enter stage left, Agile. Agile is a movement that proposes alternatives to traditional project management.[v] With its roots and typical use in software development, it is now being considered as a methodology that when adopted in a manner that emphasizes its core principles, can help businesses better respond to unpredictability. The original Agile Manifesto adheres to twelve (12) principles[vi] some of which can be extended and/or converted for a broader application to science and technology project management. Such an extension with a minor allowance for the possible need for remote communications (reference 6.) follows in summary form:

1. Priority is satisfying the customer through early and continuous delivery of *versions of the system.*
2. Welcome changing requirements, even late in development. (no change)

3. Deliver *working versions of the system* frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project. (no change)
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. (no change)
6. The most efficient and effective method of conveying information to and within a development team is *temporally synchronized, preferably* face-to-face conversation.
7. Working *(sub)system(s)* is the primary measure of progress.
8. Agile processes promote sustainable development. (no change)
9. Continuous attention to technical excellence and good design enhances agility. (no change)
10. Simplicity--the art of maximizing the amount of work not done--is essential. (no change)
11. The best architectures, requirements, and designs emerge from self-organizing teams. (no change)
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. (no change)

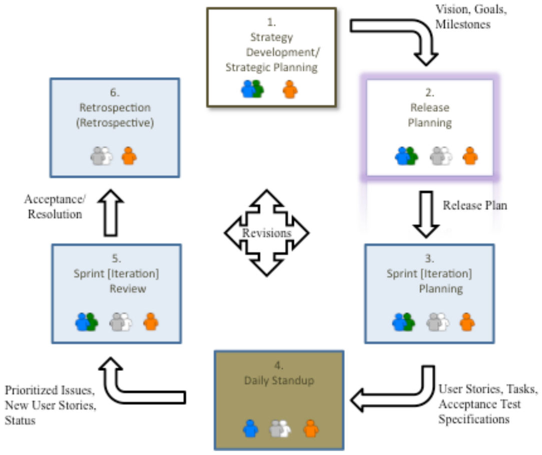
Even a cursory understanding of the broadened principles lends to a creditable argument that communicative, self-organizing, self-managing and frequently engaged teams are likely to make a major contribution to innovation in the workplace. This is a prescription for circumstance essential to successful science and technology projects.

Putting it Together

There are a number of implementations and practices of the Agile methodology that emphasize iterative development, collaboration, and accountability of each member of the team to the team itself. The list includes, but is not remotely limited to: Disciplined Agile Delivery (DAD), Dynamic Systems Development Method (DSDM) and its Alternative, Agile Unified Process (AUP), and multiple renditions of Scrum. The jewel is not in examining the detailed differences of and between the frameworks. But, rather to examine the underlying Agile principles within the context of preparation of STEM college students for the workforce. Hence, for the purpose of raising the contextual point, let's review components of "a" Scrum framework. The presented framework is not intended to reflect any pure or sanctioned modified version of Scrum.

Three Scrum roles are the Product Owner(s), Scrum Development Team and the Scrum Master. The Product Owner is responsible for the vision and is the final arbiter of the system requirements. A STEM student exposed to the conceptual process either as the visionary or interpreter of another's vision becomes familiar with the transition of an idea to a plan. They are also privy to the dynamics of competing stakeholder interests and how related issues are weighed and incorporated into the system. The Scrum Development Team is cross-functional and may include members with varying subject area expertise, implementation practitioners such as systems engineers, software developers and/or others with specialized business analysis skills. They are expected to form a self-organizing and self-managing unit without externally defined roles. [vii] STEM students engaged as a Scrum team would be afforded the opportunity to hone skills such as negotiation and collaborative productivity. Moreover, they may better appreciate the important balance struck between autonomy and accountability to the overall team. The ScrumMaster facilitates the process and addresses impediments to the team progress. Students acting in this role or observing a Scrum Master can exercise creativity in problem solving and working through others to achieve a greater good. There are also identified techniques that are associated with being a SrumMaster that a student may add to their skills portfolio that has value in other professional arenas.

For further examination of added value of exposure and practice of Agile, consider six generic activities and meetings associated with the development of a system are depicted below in a loose logical order. The Strategic Planning activity facilitates STEM student active participation in the articulation of a vision, project goals, and milestones.


1.
Strategy Development/ Strategic Planning
Vision, Goals, Milestones
2.
Release Planning
Release Plan
3.
Sprint [Iteration] Planning
User Stories, Tasks, Acceptance Test Specifications
4.
Daily Standup
Prioritized Issues, New User Stories, Status
5.
Sprint [Iteration] Review
Acceptance/ Resolution
6.
Retrospection (Retrospective)
Revisions

The development of the Release plan familiarizes STEM students with Agile components such as themes, epics, user stories, and the backlog. Sprint Planning engages the STEM student in identifying and quantifying a potentially shippable system "increment". [viii] Active participation in both the Release and Sprint Planning meetings provides insight into the interaction between all three (3) roles and negotiation between the Product

Owners and the Development Team. The Daily StandUp, which typically occurs for 15 minutes or less each day, highlights the continual accountability of each member to the team as a whole as a routine characteristic. The Sprint Review provides an opportunity for STEM students to engage in the assessment process as either the reviewer who ultimately determines if the increment is done, or as the ScrumMaster who assists the Product Owner in converting their feedback into modification of the backlog. STEM student participation in the Retrospection (or Retrospective) or reflection on the process and the overall productivity of the team may prove to be a significant experience in that it involves self-assessment of behaviors and fusion of other observations for improvement and adaptability in subsequent iterations.

Higher Educating STEM College Students

The promise of "higher educating" STEM college students through exposure to the tenets of Agile outlined above is boundless. Furthermore, herein lies an opportunity to address voids in STEM college level preparation expressed by employers. The Agile methodology may be infused and/or incorporated at different points or phases of the undergraduate and graduate matriculation. Traditional inclusion in the curriculum via course(s), electives, as part of capstone experiences, and/or through other experiential activities such as internships are all possibilities. Consistent with the spirit of Agile, the exact implementation is flexible and adaptable to the particular higher education environment.

The scope of this writing is providing a competitive advantage and workforce readiness for STEM college students. However, the idea of infusing the tenets of Agile into all college student preparation for the workforce – not just science and technology project management - holds significant promise as well.

Cited References

[i] [Staff] "Prepare the American Workforce to Compete in the Global Economy." *The Jobs Council.* Web. 29 May 2014

[ii] Ibid

[iii] Helguera, Maria and Pow, Joseph "A PROJECT-BASED PEDAGOGY FOR TECHNICAL EDUCATION AMONG FIRST YEAR COLLEGE STUDENTS." *RIT Carlson Center for Imaging Science.* Rochester Institute of Technology, 30 April 2014

[iv] Caron, Sarah. "Five Ways to Better Prepare Students for Careers." *The Education World.* Education World, 2011. Web. 30 May 2014

[v] [Staff] "Agile Methodology." AgileMethodology.org 2013. Web. June 2013

[vi] Beck, K., Beetle, M., et al. "Principles behind the Agile Manifesto" *AgileManifesto.org*, 2001. Web. March 2013

[vii] Michael, James "Scrum Reference Card" *ScrumReferenceCard,com* Web. Version 0.9I 2010-2012, March 2013.

[viii] Ibid

Highly Accurate Derivatives: A Matlab Tool for Teaching and Research

Abdulwahab A. Abokhodair
Department of Earth Sciences, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

Abstract – *Many computational tasks in science and engineering require highly accurate numerical derivatives. This paper presents a method of derivative approximation that avoids the subtractive cancellation problem that plagues ordinary finite difference (FD) and thus yields near-exact first-order derivatives. Extension of the method to second-order derivatives allows approximations with considerable reduction in round-off errors. An overview of the method and a summary of its implementation are discussed and illustrated. Numerical experiments and application examples demonstrate that the method outperforms ordinary FD schemes in accuracy, efficiency, and numerical stability. Hence, the technique has strong potential as an effective teaching and research tool in engineering and scientific computing.*

Keywords: *Numerical differentiation, derivative approximation, optimization, Hessian, Jacobian, Gradient.*

1 Introduction

The approximation of derivatives is a central and costly step in most computational tasks in science and engineering. Data inversion and parameters estimation problems, for example, require repeated computations of gradients, Jacobians and Hessians of the objective functions and associated active constraints. Inaccuracies in the approximation of these derivative structures not only degrade the performance of the optimizing algorithm but also limit the accuracy of achievable solutions. Therefore, development of efficient and accurate methods to approximate derivatives is of paramount importance.

Because of its ease of implementation, finite difference (FD) is routinely used by teachers and researchers to approximate derivatives of functions. However, FD schemes are well known to be neither efficient nor accurate. A major drawback of the FD method is its critical dependence on the differencing interval or step-size. For maximum accuracy, an optimum step size must be sought, often by trial and error, a process that significantly impedes the efficiency of the method (Mark and Workman 2003; Burge and Newman 2003). Moreover, from the perspective of filter theory, generic FD differentiation filters have the undesirable property of noise amplification (Orfanidis 1996). In the context of numerical optimization, Hong, et al. (2012), point out that use of suboptimum FD step-size in iterative procedures introduces statistical noise that adversely impact not only the convergence rate of the optimization algorithm, but more importantly the accuracy of the resulting solution as well as the solution statistical properties and confidence interval.

In this paper, I describe a method based on the theory of complex variable (hereafter referred to as complex perturbation method - CPM) as an alternative to FD with superior qualities. The technique was originally reported for first-order derivatives of univariate functions by Squire and Trapp (1998) who demonstrated that it was highly accurate, extremely robust and very easy to implement. The method has since been gaining recognition in many areas of computational fields, and has successfully been applied in several large scale studies including sensitivity analyses, aerodynamic design optimization and pseudospectral algorithms (e.g. Cerviño and Bewley, 2003; Martin et al., 2001; Burge and Newman, 2003; Vatsa, 2000; Wang, 2004; De Pauw and Vanrolleghem, 2006). Generalization of CP to multi-parameter functions, and extension to second-order derivatives were reported by the present author (Abokhodair, 2007). In addition to bringing the technique to the attention of the wider computational science community, the objective of this article is to present an overview of CPM, demonstrate by examples its accuracy and numerical stability, and illustrate its implementation in some computational problems. The aim is to highlight the simplicity, power and wide applicability of the technique, and thereby facilitate its use by students and researchers.

2 How CPM Works

A simple example illustrates how the CPM method works: Let $f(x) = x\,e^{-\sin x}$ with exact derivative $f'(x) = (1 - x\cos x)e^{-\sin x}$. To estimate $f'(x)$ by the CP method, the key idea is to perturb the target variable x with a pure imaginary step ih, $\left(i = \sqrt{-1},\ \ h \ll 1\right)$, and construct the complex argument $z = x + ih$. Thus, the 'complexified' version of the original real-valued function becomes:

$$f\,(z) = z\;e^{-\sin z}\ .$$

For h << 1,we may write:

$\sin z\ = \sin\left(x + ih\right) \cong \sin x + ih\cos x$,

therefore, $e^{-\sin z}\ = e^{-\sin x}e^{-i\alpha}$, where $\alpha = h\cos x$.

Now $e^{-\mathrm{i}\alpha} = \cos\alpha - i\,\sin\alpha \cong 1 - ih\cos x$,

$e^{-\sin z}\ = \left(1 - ih\cos x\,\right)e^{-\sin x}$, and,

$f\,(z\,) \cong \left(x + ih\right)\left(1 - ih\cos x\,\right)e^{-\sin x}$. Finally we have:

$$f\left(z\right)\cong xe^{-\sin x}+ih\left(1-x\cos x\right)e^{-\sin x}$$

This is a remarkable result. The complexified function $f(z)$ returns $O(h^2)$ approximations of the original real-valued function and of its first derivative in the real and imaginary parts respectively. Moreover, the first derivative returned in the imaginary part of $f(z)$ involves no differencing operation, thus precluding any subtractive cancellation and allowing for near-exact approximations. This is the essence of the CP method. Note also how the complex data type serves here as a carrier of both the function approximation in the real part and the derivative approximation in the imaginary part. This closely resembles true automatic differentiation systems implemented in object-oriented programming. The result above is general as may be proven by a Taylor expansion of $f(z)$: $f(z)=f(x+ih)\approx f(x)+ihf'(x)+O(h^2)$, which is the original formula reported by Squire and Trapp (1998).

Figure 1 compares the absolute relative errors of the CPM and centered finite difference (CFD) approximations of $f'(z)$ computed with respective step sizes of h= 10^{-25} and 10^{-8} As seen in the figure, the CPM relative error fluctuates randomly near machine epsilon (10^{-16}), whereas the CFD error is eight orders of magnitude larger, with a mean value of about 10^{-8}. The Matlab script in box 1, computes the CPM derivative of this example.

```
x=pi*(-1:.02:1)';
h=1.0e-25 ;
f=@(x) x.*exp(-sin(x));
z=complex(x,h);
fc=f(z);
cpfp=imag(fc)/h;
```

Box 1: Matlab script to compute f(x) and f'(x) approximations of the heuristic example.

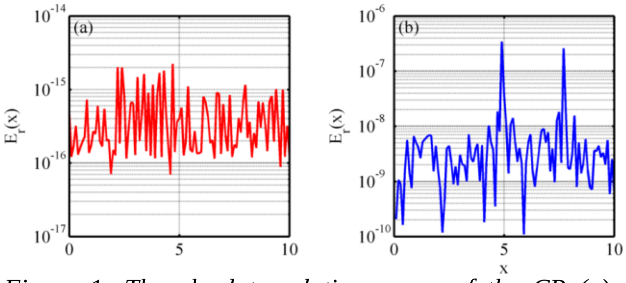


Figure 1: The absolute relative errors of the CP (a), and CFD (b) approximations of $f'(x)$ for the heuristic example using step sizes h = 10^{-25} and 10^{-8} respectively.

3 Computing Jacobians

The generalized CPM formula for vector-valued functions of several variables is:

$$\mathbf{F}(\mathbf{z})=\mathbf{F}(\mathbf{x}+ih\mathbf{e})=\mathbf{F}(\mathbf{x})+ih\mathbf{e}^T\mathbf{J}(\mathbf{x})+O\left(h^2\right), \qquad (1)$$

where $\mathbf{F}(\mathbf{x})=\left[f_k(\mathbf{x})\right]_{k=1}^{N}$ $\mathbf{x}=\left[x_k\right]_{k=1}^{M}$, and the $N\times M$ matrix $\mathbf{J}(\mathbf{x})$ is the Jacobian with respect to **x**, (Abokhodair, 2007). The imaginary part of equation (1) which provides the approximation formula for first derivatives is:

$$\mathbf{e}^T\mathbf{J}(\mathbf{x})=\frac{1}{h}Im\left[\mathbf{F}(\mathbf{z})\right]+O\left(h^2\right) \qquad (2)$$

The Jacobian (gradients are special cases) is a first-order derivative structure extensively used in scientific computing in various contexts. Implementation of equation 2 is illustrated for the Jacobian of the environmental model in equation 3, which is a standard test function used for parameters calibration and tuning. The function models a pollutant spill caused by a chemical accident and returns the concentration of the pollutant at the space-time vector $\mathbf{x}=\left(x,t\right)$, where $0\le x\le 3$ and $t>0$ (Bliznioukt, et al. 2008)

$$\mathbf{F}(\mathbf{p},\mathbf{x})=\frac{M}{\sqrt{Dt}}\exp\left(\frac{-x^2}{4Dt}\right)+\frac{M}{\sqrt{D(t-\tau)}}\exp\left(\frac{-(x-L)^2}{4D(t-\tau)}\right) \qquad (3)$$

$\mathbf{p}=\left[M,D,L,\tau\right]$ is the vector of parameters: M = mass of pollutant; D = diffusion rate; L,τ = location and time of the second spill. Figure 2 depicts the residual errors in the FD-based Jacobian compared to the CP Jacobian. As seen in the figure, the residuals are relatively large (> 10^{-6}) and systematic in character. Accumulation of such errors in iterative procedure can contribute to the ill-conditioning of the computed Jacobian leading at best to biased solutions or at worst to failure of the computational routine. To appreciate the significance of the errors indicated in figure 3, suppose that the computed Jacobian $\tilde{\mathbf{J}}$ is: $\tilde{\mathbf{J}}=\mathbf{J}+\delta\mathbf{J}$, with an error of $\delta\mathbf{J}$, and it is entered into a computation of the form $\mathbf{y}=\mathbf{J}\,\mathbf{x}$, then an upper bound on the relative error in the solution $\mathbf{x}$ is (e.g. Treferthan, and Bau, 1997) :

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}\le\kappa(\mathbf{J})\frac{\|\delta\mathbf{J}\|}{\|\mathbf{J}\|},$$

which in the present case is not negligible being of the order 10^{-2}.

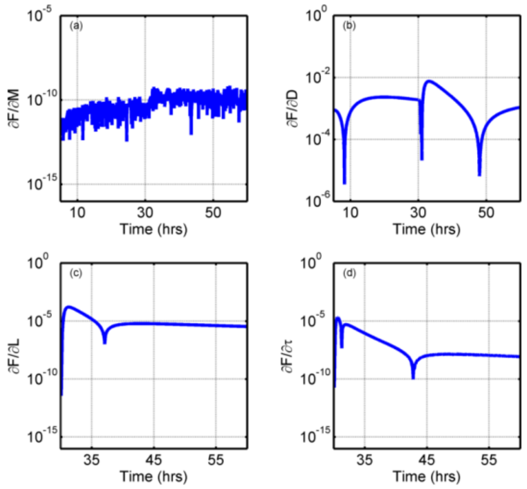


Figure 2: Residual errors in the Jacobian matrix of the environmental function in equation 3

4 Computing Hessians

The approximation formula for first order derivative (equation 1) is the only subtraction-free formula that can be obtained from complex variable theory. However, at the expense of a small lose in accuracy this formula can be extended to second-order derivatives by any ordinary finite difference scheme. This may be accomplished by perturbing the target variable say x, by a full complex step $\delta z = \delta x + ih$ and then expanding in Taylor series. The approximation formula so obtained using a centered difference (CFD) scheme is (Abokhodair, 2007):

$$\mathbf{e}^T\mathbf{H}\,\mathbf{e} = \frac{1}{2h\delta x}\mathrm{Im}\left[\Delta f(\mathbf{z})\right] + O(h^2) \qquad (4)$$

where $\mathbf{H}$ is the hessian matrix.

The hessian matrix, a second-order derivative structure, plays a crucial role in optimization algorithms allowing for the quadratic approximation of the objective function locally. To demonstrate application of equation 4, the Colville function is used, which is a standard optimization test function defined as (Surjanovic and Bingham, 2013):

$$\begin{aligned} f(\mathbf{x}) &= g_1(\mathbf{x}) + g_2(\mathbf{x}) \\ g_1(\mathbf{x}) &= 100\left(x_1^2 + x_2\right)^2 + \left(x_1 - 1\right)^2 + \left(x_3 - 1\right)^2 + 90\left(x_3^2 + x_4\right)^2 . \\ g_2(\mathbf{x}) &= 10.1\left[\left(x_2 - 1\right)^2 + \left(x_4 - 1\right)^2\right] + 19.8\left(x_2 - 1\right)\left(x_4 - 1\right) \end{aligned}$$

The function is usually evaluated on the hypercube: $[x_i]_{i=1}^4 \in [-10, 10]$, and has a global minimum: $f(\hat{\mathbf{x}}) = 0$, at $\hat{\mathbf{x}} = (1,1,1,1)$. Table 1 compares the element-wise relative errors in the Hessians of the objective near its minimum value computed by the CPM and CFD schemes. As seen in the table, the CPM-based Hessian is 7 orders of magnitude more accurate than that of the CFD-based approximation. In terms of norms, the relative errors are of the orders: 10^{-12} and 10^{-6}, respectively a substantial gain in accuracy by the CPM scheme (Eqution 4) despite the fact that it involves a differencing operation.

Table 1: Relative errors in the Hessian of Colville test function (Eqution 5)

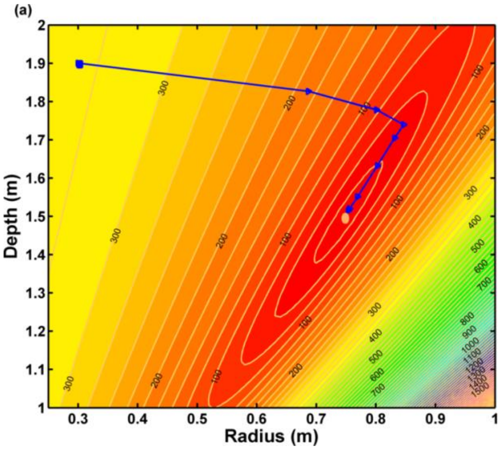
CPM Hessian × 10^{-10}			
0.0655	0.0655	0	0
0.0655	0.1874	0	0.0655
0	0	0.1902	0.0655
0	0.0655	0.0655	0.0655

CFD Hessian × 10^{-3}			
0	0	0.2032	0.0508
0	0	0.4064	0
0.2032	0.4064	0	0
0.5080	0	0	0

5 Application Examples

Use of the CP method in computational problems is demonstrated with an optimization example. In near-surface geophysical applications (e.g. archaeology and environmental studies), ground magnetics is often the method of choice for locating small buried metallic objects such as archaeological artifacts, drums and unexploded ordnances (UXOs). For this purpose, an isolated dipole or, equivalently, a uniformly magnetized spherical source is an adequate model of the anomaly source. Assuming induced magnetization, the total-field magnetic anomaly of a sphere of radius R and magnetization J buried at depth zo is given by (e.g. Blakely, 1995):

$$\begin{aligned} \Delta B(\mathbf{p}, x) &= \frac{KR^3}{r^5}\left(A x^2 + z_o B x + C z_o^2\right) \qquad (5) \\ A = 3\cos^2(I) - 1 \quad & B = -3\sin(2I) \quad C = 3\sin^2(I) - 1 \end{aligned}$$



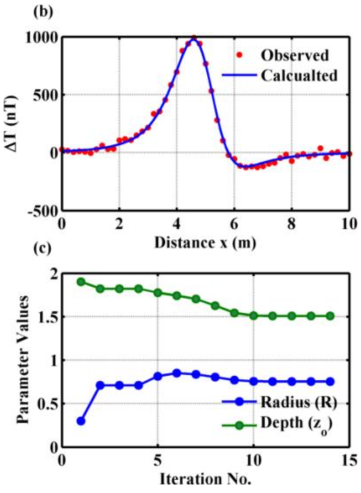


Figure 3: Results of the inversion example - (a) contours of the objective function and superimposed convergence path (blue); (b) observed data and solution-based model response; (c) iteration paths of the two parameters.

In equation 5, $\mathbf{p} = [R, z_o]$ is the vector of parameters, *I* is the geomagnetic inclination and *K* is a constant. For this didactic example, synthetic data was generate from equation 6 for a source of radius R = 0.75 m at depth z_o = 1.5 m. A normal random error of zero mean and a standard deviation 2.5% of the peak anomaly was added to simulate observational data. The inverse problem solved consisted in estimating the parameters *R* and z_o from the noisy data. The inversion results (figure 3) are: *R* = 0.75002 ± 0.00071 and z_o = 1.50137 ± 0.0038. The inversion algorithm used is based on Newton method with a Levenberg-Marquardt type damping and requires at each iteration step updated versions of the Hessian and gradient of the objective function and associated constraints. The derivative structures were supplied by a Matlab function that was integrated into the optimizing routine.

6 Summary

The CP method introduced in this paper is a hybrid between finite difference (FD) and true automatic differentiation (AD). Computationally, it is simply FD in the complex plane, but without the step-size dilemma. And, in terms of performance, it is competitive with AD. The examples presented here demonstrate some of the key advantages of the CP technique: namely, implementation simplicity, accuracy, and numerical stability throughout a wide range of step-sizes down to 10^{-50}. Because of these performance features, the method provides a better alternative to ordinary FD schemes as on-the-fly automatic differentiation tool already built into most common computational environments.

7 Acknowledgements

This research work was conducted at King Fahd University of Petroleum and Minerals, (KFUPM), Department of Earth Sciences, Dhahran, Saudi Arabia. Support of KFUPM is gratefully acknowledged. I also thank Professor A. Dogan and Dr. K. Al-Ramadan for reviewing the manuscript and for their valuable suggestions.

8 References

[1] Abokhodair, A. A.: Numerical tools for geoscience computations: Semiautomatic differentiation—SD. Comput. Geosci. 11, 283-296 (2007).

[2] Burg, C. O. E, and Newman, J.C.: Computationally efficient, numerically exact design space derivatives via the complex Taylor's series expansion method. Computers & Fluids 32, 373-383 (2003).

[3] Cerviño, L. I. and Bewley,T. R.: On the extension of the complex-step derivative technique to pseudospectral algorithms, J. Comp. Phys., 187, 544-549 (2003).

[4] Mark, H. and Workman Jr., J.: Derivatives in Spectroscopy: Part I – The Behavior of the derivative, Spectroscopy 18(4), 32-37 (2003).

[5] Martin, J. R. R. A., Kroo, I. M. and Alonso, J. j.: An Automated method for sensitivity analysis using complex variables, Paper AIAA 2001-689 (2001).

[6] Squire, W. and Trapp, G. Using complex variables to estimate derivatives of real functions, SIAM Rev., 40(1), 110-112 (1998).

[7] Vatsa, V.: Computation of sensitivity derivatives of Navier-Stokes equations using complex variables. Advances in Engineering Software 31, 655–659 (2000).

[8] Wang, B. P.: Complex variable method for sensitivity analysis in structural optimization. 4th World Congress on Computational Mechanics, in conjunction with APCOM (2004).

[9] DePauw, D. J. W. and Vanrolleghem, P. A.: Avoiding the finite difference sensitivity analysis deathtrap by using the complex-step derivative approximation technique. 3rd biennial meeting of the International Environmental Modeling and Software Society (iEMSs), Burlington (2006).

[10] Orfanidis, S.J.: Introduction to Signal Processing. Prentice Hall, Englewood Cliffs, NJ, (1996).

[11] Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved April 5, 2014, from http://www.sfu.ca/~ssurjano.

SESSION

POSTER PAPER

Chair(s)

TBA

Collaborative Research Project for CS and Engineering Students

Subodh Bhandari, Fang Tang, Zekeriya Aliyazicioglu, Scott Boskovich and Kevin La
California State Polytechnic University, Pomona
Pomona, CA, 91768
Email: {sbhandari|ftang|zaliyazici|sboskovich|kyla}@csupomona.edu

***Abstract*—This poster paper describes a multi-disciplinary and multi-year research project that involves students from Computer Science (CS), Aerospace Engineering (ARO), Electrical and Computer Engineering (ECE) and Mechanical Engineering (ME). They work as a team on solving the research problem of collaboration between Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs).** [1]

I. The Project

The problem that we are interested in is the collaboration between UAVs and UGVs to accomplish search and rescue tasks. Figure 1 describes one of the task scenario, where one UAV is flying in patterns to search a particular area for a potential object of interest. Once the geo-location of the target is found, it is transmitted over the network to the other two UAVs, which will fly near the target and drop water bottles to that location for rescue purposes. To make sure that multiple UAVs can fly at the same time, UAVS are programmed with the collision avoidance capabilities. Meanwhile, a UGV is set to navigate to the target location and verify the delivery. This project is naturally a multi-disciplinary project which requires the knowledge from ARO students for flight control, ECE students for the hardware design, ME students for mechanics, and CS students for software control.

This project has been sponsored by the Northrop Grumman Corporation (NGC) since 2011. The Pomona team also collaborated with the Cal Poly Slo team to divide and conquer the task. To demonstrate our project progress, we perform a yearly midterm design review to NGC and a final demonstration of the integrated system to NGC and the public every year. We are currently in year three of the project. This poster paper mainly describes the research project done in the Pomona team for the current year.

II. The Approach

A. Key Research Issues

1) UAV Collision Avoidance: The key operations required by our UAV includes: autonomous waypoint navigation, collision avoidance, autonomous payload release (water bottle), and manual override and failsafe (if needed). Our focus this year handles collision avoidance. We first perform simulation experiments to prove the avoidance capability. The mass properties of both UAVs are modeled and provided to the simulation team. The collision avoidance is essentially a detect-sense-avoid process. During the flight, UAVs will communicate their status such as position and velocity. Due to the unreliable GPS data, a kalman filter is used to approximate the UAV's position. When there's a potential collision, waypoints will be generated to steer UAV away from the collision course.

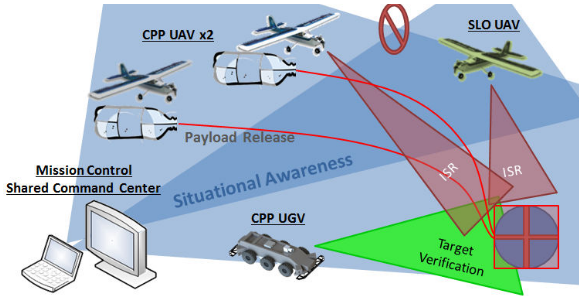


Fig. 1. Our task scenario with three UAVs and one UGV.

2) UGV Navigation: The tasks of the UGV include autonomous target search and recognition, retreat to safe zone, and return to verify the package delivery on target. The UGV uses a FPGA interface for motor control, sensor data acquisition (GPS, IMU, Sonars), and power control. There's an onboard processor to handle camera and lidar data and control the behavior layer. To achieve the prescribed tasks, the UGV is programmed with the following capabilities: 1) target recognition using OpenCV2. The primary target's position is calculated using current robot's state; 2) path planning using D* Lite [1]; and 3) waypoint navigation, including obstacle avoidance.

3) Ground Control Station (GCS): The GCS provides a graphical user interface for the human operator to oversee the team operation and intervene at critical time. Our goal is to maximize situation awareness and minimize the cognitive load of the operator. Through trade studies, we selected the following tools: the Qt framework, which provides user interface and database access; the ArcGIS, a geographical information system software which has the ability to show the map, add overlays, and allows users to interact with the map (zooming and panning); SQLite database; and our own Protonet communication protocol.

4) Communication: In our system, the UAVs need to communicate directly with each other for collision avoidance data. UAVs and UGVs also send their status back to the GCS. GCS may send command and control to the vehicles. We

[1] This paper is submitted as a poster.

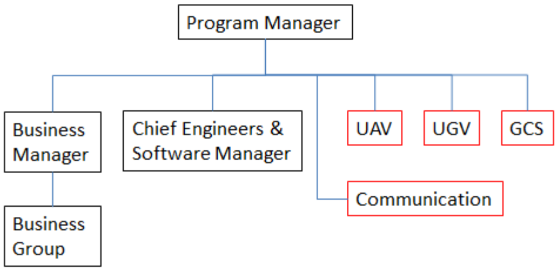


Fig. 2. Our team organization.

designed and developed our own Protonet protocol, which is a lightweight protocol that supports multiple platforms and languages. This is important because we have a large team with different expertise and we do not want to limit to one system and one language for software development.

B. Team Management

We have actively recruited students from CS, ARO, ECE and ME to work together in a large team. Each department also has one advisor to guide the research. Our student participation has increased significantly from 25 students in the first year, to 50 and 75 in the subsequent years. With such a large group of students, we need to have a management plan for the group to work efficiently towards the goal. We adopted a hierarchical management structure. The student body is led by a program manager with a support group of business majors for handling daily team operations, budget, logistics, and recruitment. Technically, the group is further divided into a group of chief engineers and software manager who have more expertise of the overall system, and a group for each key research area including UAV, UGV, GCS and communication. Each group is appointed a team lead and a business major to organize the research and operation within the group. Figure 2 gives an overview of our organization.

This year, our student body is mainly composed of undergraduate students and with 9 percent of graduate students. We also recognize that we need to have seats for Freshman or Sophomore students, since the seniors will graduate at the end of the year. We emphasize on mentoring and training of young students, encouraging young leaders, and promoting women in technical and leadership positions. As you can see from Figure 3, we have about 21 percent of returning members this year. Most of them are serving as mentors and leadership roles. The percentage of women participants also increased from 8 percent in the first year to 14 percent and 23 percent in the second and third year.

III. Results

A. Technical Results

Over the years, we have already successfully demonstrated our integrated system. In year one, our Pomona team mainly focused on the design and control of the UGV. We have successfully built our own UGV with on board computing, laser range finder, camera, IMU, sonars and FPGAs for sensor processing. The UGV has carried out the mission of waypoints navigation and obstacle avoidance. In year two, our team focused on developing the dropping mechanics and the dropping algorithm on a Sig Kadet airframe. We also optimized our hardware design for the UGV and added functionality such as target recognition. In our year-end demonstration, the UAV had an autonomous flight control and successfully dropped the water bottle near the target location. This year, we will demonstrate the more challenging UAV collision avoidance behavior in May 2014.

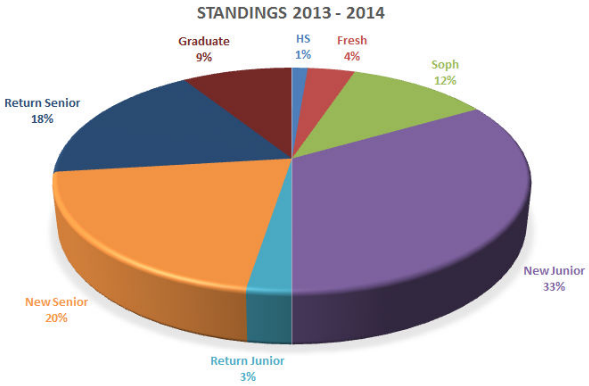


Fig. 3. Our current standings for 2013-14.

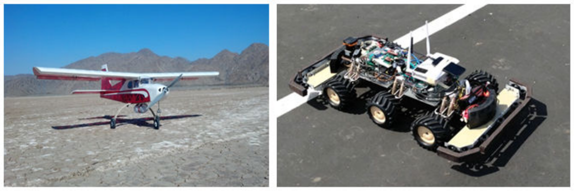


Fig. 4. The Sig Kadet (left) and the UGV (right).

B. Other Benefits

Through a year's training, students from different departments get to work together as a team on a real-world application, which helps them to be a good team player, and to learn things that are not directly taught in the classroom. They all go through the whole process of design, development, and delivery of the product. Students gained more experience on their presentation skills and problem solving skills. The CS students have the opportunity to learn more on hardware and the engineering students can learn more about software. Many students evolved as team leads. Students also build their industrial connections with NGC. Several of our students got internship and full-time offers at NGC after their completion of the project.

IV. Conclusion

As a conclusion, this multi-disciplinary project has prepared the students with more hands-on experience, which is good for their future career development.

References

[1] S. Koenig and M. Likhachev. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics and Automation*, 2005.